From low probability to high confidence in stochastic convex optimization

Damek Davis* Dmitriy Drusvyatskiy† Lin Xiao‡ Junyu Zhang§

Abstract

Standard results in stochastic convex optimization bound the number of samples that an algorithm needs to generate a point with small function value in expectation. More nuanced *high probability* guarantees are rare, and typically either rely on "light-tail" noise assumptions or exhibit worse sample complexity. In this work, we show that a wide class of stochastic optimization algorithms for strongly convex problems can be augmented with high confidence bounds at an overhead cost that is only logarithmic in the confidence level and polylogarithmic in the condition number. The procedure we propose, called `proxBoost`, is elementary and builds on two well-known ingredients: robust distance estimation and the proximal point method. We discuss consequences for both streaming (online) algorithms and offline algorithms based on empirical risk minimization.

1 Introduction

Stochastic convex optimization lies at the core of modern statistical and machine learning. Standard results in the subject bound the number of samples that an algorithm needs to generate a point with small function value in *expectation*. Specifically, consider

$$\min_x \; f(x) := \mathbb{E}_{z\sim\mathcal{P}}[f(x,z)], \tag{1.1}$$

where the random variable z follows a fixed unknown distribution $\mathcal{P}$ and $f(\cdot,z)$ is convex for almost every $z\sim\mathcal{P}$. Given a small tolerance $\epsilon>0$, stochastic gradient methods typically produce a point x_ϵ satisfying

$$\mathbb{E}[f(x_\epsilon)] - \min f \le \epsilon.$$

*School of ORIE, Cornell University, Ithaca, NY 14850, USA; `people.orie.cornell.edu/dsd95/`.

†Department of Mathematics, U. Washington, Seattle, WA 98195; Microsoft Research, Redmond, WA 98052; `www.math.washington.edu/~ddrusv`. Research of Drusvyatskiy was supported by the NSF DMS 1651851 and CCF 1740551 awards.

‡Microsoft Research, Redmond, WA, USA; `www.microsoft.com/en-us/research/people/lixiao/`

§Department of Industrial and Systems Engineering, University of Minnesota, Minneapolis, MN, USA; `zhan4393@umn.edu`

The cost of the algorithms, measured by the required number of stochastic (sub-)gradient evaluations, is $\mathcal{O}(1/\epsilon^2)$ or $\mathcal{O}(1/\epsilon)$ if f is strongly convex (e.g., [18, 35, 39]).

In this paper, we are interested in procedures that can produce an approximate solution with *high probability*, meaning a point $x_{\epsilon,p}$ satisfying

$$\mathbb{P}(f(x_{\epsilon,p}) - \min f \leq \epsilon) \geq 1 - p, \tag{1.2}$$

where $p > 0$ can be arbitrarily small. By Markov's inequality, one can guarantee (1.2) by generating a point $x_{\epsilon,p}$ satisfying $\mathbb{E}[f(x_{\epsilon,p})] - f^* \leq p\epsilon$, e.g., by using standard stochastic gradient methods. However, the resulting sample complexity can be very high for small p with the typical scaling of $\mathcal{O}(1/(p\epsilon))$ or $\mathcal{O}(1/(p\epsilon)^2)$. Existing literature does provide a path to reducing the dependence of the sample complexity on p to $\log(1/p)$, but this usually comes with cost of either worse dependence on ϵ (e.g., [8,38,42]) or more restrictive sub-Gaussian assumptions on the stochastic gradient noise (e.g. [17, 18, 24, 34]).

We aim to develop *generic* low-cost procedures that equip stochastic optimization algorithms with high confidence guarantees, without making restrictive noise assumptions. Consequently, it will be convenient to treat such algorithms as black boxes. More formally, suppose that the function f may only be accessed through a *minimization oracle* $\mathcal{M}(f, \epsilon)$, which on input $\epsilon > 0$, returns a point x_ϵ satisfying the low confidence bound

$$\mathbb{P}(f(x_\epsilon) - \min f \leq \epsilon) \geq \frac{2}{3}. \tag{1.3}$$

(By Markov's inequality, minimization oracles arise from any algorithm that can generate x_ϵ satisfying $\mathbb{E}f(x_\epsilon) - \min f \leq \epsilon/3$.) Let $\mathcal{C}_{\mathcal{M}}(f, \epsilon)$ denote the cost of the oracle call $\mathcal{M}(f, \epsilon)$. Given a minimization oracle and its cost, we investigate the following question:

> Is there a procedure within this oracle model of computation that returns a point $x_{\epsilon,p}$ satisfying the high confidence bound (1.2) at a total cost that is only a "small" multiple of $\mathcal{C}_{\mathcal{M}}(f, \epsilon) \cdot \log(\frac{1}{p})$?

We will see that when f is strongly convex, the answer is yes for a wide class of oracles $\mathcal{M}(f, \epsilon)$. To simplify discussion, suppose f is μ-strongly convex and L-smooth (differentiable with L-Lipschitz continuous gradient). Then the cost $\mathcal{C}_{\mathcal{M}}(f, \epsilon)$ typically depends on the condition number $\kappa := L/\mu \gg 1$, as well as scale sensitive quantities such as initialization quality and upper bound on the gradient variances, etc. The procedures introduced in this paper execute the minimization oracle multiple times in order to boost its confidence, with the total cost on the order of

$$\log\left(\frac{\log(\kappa)}{p}\right) \log(\kappa) \cdot \mathcal{C}_{\mathcal{M}}\left(f, \tfrac{\epsilon}{\log(\kappa)}\right).$$

Thus, high probability bounds are achieved with a small cost increase, which depends only logarithmically on $1/p$ and polylogarithmically on the condition number κ.

Before introducing our approach, we discuss two techniques for boosting the confidence of a minimization oracle, both of which have limitations. As a first approach, one may query the oracle $\mathcal{M}(f,\epsilon)$ multiple times and pick the "best" iterate from the batch. This is a flawed strategy since often one cannot test which iterate is "best" without increasing sample complexity. To illustrate, consider estimating the expectation $f(x) = \mathbb{E}_z[f(x,z)]$ to ϵ-accuracy for a fixed point x. This task amounts to mean estimation, which requires on the order of $1/\epsilon^2$ samples, even under sub-Gaussian assumptions [10]. In this paper, the cost $\mathcal{C}_{\mathcal{M}}(f,\epsilon)$ typically scales at worst as $1/\epsilon$, and therefore mean estimation would significantly degrade the overall sample complexity.

The second approach leverages the fact that, with strong convexity, (1.3) implies

$$\mathbb{P}(\|x_\epsilon - \bar{x}\| \leq \sqrt{2\epsilon/\mu}) \geq \frac{2}{3},$$

where $\bar{x}$ is the minimizer of f. Given this bound, one may apply the *robust distance estimation* technique of [35, p. 243] and [20] to choose a point near $\bar{x}$: Run m trials of $\mathcal{M}(f,\epsilon)$ and find one iterate x_{i^*} around which the other points "cluster". Then the point x_{i^*} will be within a distance of $\sqrt{18\epsilon/\mu}$ from $\bar{x}$ with probability $1 - \exp(-m/18)$. The downside of this strategy is that when converting naively back to function values, the suboptimality gap becomes $f(x_{i^*}) - \min f \leq \frac{L}{2}\|x_{i^*} - \bar{x}\|^2 \leq 9\kappa\epsilon$. Thus the function gap at x_{i^*} may be significantly larger than the expected function gap at x_ϵ, by a factor of the condition number. Therefore, robust distance estimation exhibits a trade-off between robustness and efficiency.

The robustness/efficiency trade-off disappears for perfectly conditioned losses. Therefore, it appears plausible that one might avoid the κ factor through a continuation procedure that solves a sequence of nearby, better conditioned problems. This is the strategy we explore here. The `proxBoost` procedure embeds robust distance estimation inside a proximal point method. It begins by declaring the initial point x_0 to be the output of the robust distance estimator for minimizing f. Then the better conditioned function

$$f^t(x) := f(x) + \frac{\mu 2^t}{2}\|x - x_t\|^2,$$

is formed and the next iterate x_{t+1} is declared to be the output of the robust distance estimator for minimizing f^t. Since the conditioning of f^t rapidly improves with t, the robust distance estimator becomes more efficient as the counter t grows.

The `proxBoost` method can be applied to a wide class of stochastic minimization oracles, including both streaming algorithms (e.g., stochastic gradient methods) and offline methods such as empirical risk minimization (ERM). We now illustrate the consequences of `proxBoost` for solving the problem (1.1) using these two types of oracles.

1.1 Streaming Oracles

Stochastic gradient methods can be treated as minimization oracles $\mathcal{M}(f,\epsilon)$ whose cost $\mathcal{C}_{\mathcal{M}}(f,\epsilon)$ are measured by the number stochastic gradients needed to reach functional accuracy ϵ in expectation. An algorithm with minimal such cost was proposed by Ghadimi and Lan [18]. It generates a point x_ϵ satisfying $\mathbb{E}[f(x_\epsilon) - \min f] \leq \epsilon$ with

$$\mathcal{O}\left(\sqrt{\kappa}\ln\left(\frac{\Delta_{\text{in}}}{\epsilon}\right) + \frac{\sigma^2}{\mu\epsilon}\right) \tag{1.4}$$

stochastic gradient evaluations, where the quantity σ^2 is an upper bound on the variance of the stochastic gradient estimator $\nabla f(x,z)$ and Δ_{in} is a known upper bound on the initial function gap $\Delta_{\text{in}} \geq f(x_0) - f^*$. A simpler algorithm with a similar efficiency estimate was recently presented by Kulunchakov and Mairal [25], and was based on estimate sequences. Aybat et al. [4] developed an algorithm with similar efficiency, but in contrast to previous work, it does not require the variance σ^2 and the initial gap Δ_{in} as inputs.

It is intriguing to ask if one can equip the stochastic gradient method and its accelerated variant with high confidence guarantees. In their original work [17,18], Ghadimi and Lan provide an affirmative answer under the additional assumption that the stochastic gradient estimator has light tails. The very recent work of Juditsky-Nazin-Nemirovsky-Tsybakov [23] shows that one can avoid the light tail assumption for the basic stochastic gradient method, and for mirror descent more generally, by truncating the gradient estimators. High confidence bounds for the accelerated method, without light tail assumptions, remain open.

In this work, the optimal method of [18] will be used as a minimization oracle within `proxBoost`, allowing us to nearly match the efficiency estimate (1.4) without "light-tail" assumptions. Equipped with this oracle, `proxBoost` returns a point $x_{\epsilon,p}$ satisfying (1.2) and the overall cost of the procedure is

$$\widetilde{\mathcal{O}}\left(\log\left(\frac{1}{p}\right)\left(\sqrt{\kappa}\ln\left(\frac{\Delta_{\text{in}}}{\epsilon}\vee\kappa\right)+\frac{\sigma^2}{\mu\epsilon}\right)\right).$$

Here, $\widetilde{\mathcal{O}}(\cdot)$ only suppresses logarithmic dependencies in κ; see Section 5 for a precise guarantee. Thus for small ϵ, the sample complexity of the robust procedure is roughly $\log(1/p)$ times the efficiency estimate (1.4) of the low-confidence algorithm.

1.2 Empirical Risk Minimization Oracles

An alternative approach to streaming algorithms, such as the stochastic gradient method, is based on empirical risk minimization (ERM) or sample average approximation (SAA) [44]. Namely, we draw i.i.d. samples $z_1, \ldots, z_n \sim \mathcal{P}$ and minimize the empirical average

$$\min_x \; f_S(x) := \frac{1}{n}\sum_{i=1}^{n} f(x, z_i). \tag{1.5}$$

A key question is to determine the number n of samples that would ensure that the minimizer x_S of the empirical risk f_S has low generalization error $f(x_S) - \min f$, with reasonably high probability. There is a vast literature on this subject; see for example [5,20,41,42]. We build here on the work of Hsu-Sabato [20], who focused on high confidence guarantees for nonnegative losses $f(x,z)$. They showed that the empirical risk minimizer x_S yields a robust distance estimator of the true minimizer of f. As a consequence they deduced that ERM can find a point x_S satisfying the relative error guarantee

$$\mathbb{P}\big[f(x_S) \leq (1+\gamma)f^*\big] \geq 1 - p,$$

with the sample complexity n on the order of

$$\mathcal{O}\left(\log\left(\frac{1}{p}\right)\cdot\frac{\hat{\kappa}\,\kappa}{\gamma}\right).$$

Loosely speaking, here κ and $\hat{\kappa}$ are the condition numbers of f and f_S, respectively. By embedding ERM within `proxBoost`, we obtain the much better sample complexity

$$\widetilde{\mathcal{O}}\left(\log\left(\frac{1}{p}\right)\left(\frac{\hat{\kappa}}{\gamma}+\hat{\kappa}\right)\right),$$

where the symbol $\widetilde{\mathcal{O}}$ only suppresses polylogarithmic dependence on κ and $\hat{\kappa}$. See Section 4 for the precise sample complexity guarantee.

1.3 Convex composite optimization

The results, previewed so far rely on the assumption that f is strongly convex and smooth. These techniques can not directly accommodate constraints or nonsmooth regularizers. To illustrate the difficulty, consider the convex composite optimization problem

$$\min_x f(x) = g(x) + h(x), \tag{1.6}$$

where $g\colon \mathbf{R}^d \to \mathbf{R}$ is smooth and strongly convex and $h\colon \mathbf{R}^d \to \mathbf{R}\cup\{+\infty\}$ is an arbitrary closed convex function. For example, a constrained optimization problem can be modeled by setting h to be zero on the feasible region and plus infinity elsewhere. The approach for the unconstrained problems, outlined previously, heavily relies on the fact that the function gap $f(x)-\min f$ and the squared distance to the solution $\|x-\bar{x}\|^2$ are proportional up to multiplication by the condition number. The analogous statement for the composite setting (1.6) is decisively false. In particular, it is unclear how to turn low probability guarantees on the function gap $f(x) - \min f$ to high probability outcomes, even if one was willing to degrade the accuracy by the condition number of g.

In the last section of the paper, we resolve this apparent difficulty and thereby generalize the `proxBoost` framework to the entire composite problem class (1.6). The key tool is a new robust distance estimation technique for convex composite problems, which may be of independent interest. Consequences for regularized empirical risk minimization and proximal streaming algorithms, in the spirit of Sections 1.1 and 1.2, follow immediately.

1.4 Related literature

Our paper rests on two pillars: the proximal point method and robust distance estimation. The two techniques have been well studied in the optimization and statistics literature respectively. The proximal point method was introduced by Martinet [30, 31] and further popularized by Rockafellar [40]. This construction is also closely related to the smoothing function of Moreau [33]. Recently, there has been a renewed interest in the proximal point method, most notably due to its uses in accelerating variance-reduction methods for minimizing finite sums of convex functions [15,26,27,43]. The proximal point method has also featured prominently as a guiding principle in nonconvex optimization, with the works of [2,3,12–14]. The stepsize schedule we use within the proximal point method is geometrically decaying, in contrast to the more conventional polynomially decaying schemes. Geometrically decaying schedules for subgradient methods were first used by Goffin [19] and have regained some attention recently due to their close connection to the popular step-decay schedule in stochastic optimization [4,16,45,46].

Robust distance estimation has a long history. The estimator we use was first introduced in [35, p. 243], and can be viewed as a multivariate generalization of the median of means estimator [1,21]. Robust distance estimation was further investigated in [20] with a focus on high probability guarantees for empirical risk minimization. A different generalization based on the geometric median was studied in [32]. Other recent articles related to the subject include median of means tournaments [28], robust multivariate mean estimators [22,29], and bandits with heavy tails [9].

One of the main applications of our techniques is to streaming algorithms. Most currently available results that establish high confidence convergence guarantees make sub-Gaussian assumptions on the stochastic gradient estimator [17,18,24,34]. More recently, there has been renewed interest in obtaining robust guarantees without the light-tails assumption. For example, the two works [11,47] make use of the geometric median of means technique to robustly estimate the gradient in distributed optimization. A different technique was recently developed by Juditsky et al. [23], where the authors establish high confidence guarantees for mirror descent type algorithms by truncating the gradient.

The outline of the paper is as follows. Section 2 presents the problem setting and robust distance estimation. Section 3 develops the `proxBoost` procedure. Section 4 presents consequences for empirical risk minimization, while Section 5 discusses consequences for streaming algorithms, both in the strongly convex and smooth setting. The final Section 6 extends the aforementioned techniques to convex composite problems.

2 Problem setting

Throughout, we follow standard notation of convex optimization, as set out for example in the monographs [6,37]. We let $\mathbf{R}^d$ denote an Euclidean space with inner product $\langle\cdot,\cdot\rangle$ and the induced norm $\|x\| = \sqrt{\langle x, x\rangle}$. The symbol $B_\varepsilon(x)$ will stand for the closed ball around x of radius $\varepsilon > 0$. We will use the shorthand interval notation $[1, m] := \{1, \ldots, m\}$ for any number $m \in \mathbb{N}$. Abusing notation slightly, for any set of real numbers $\{r_i\}_{i=1}^m$ we will let $\texttt{median}(r_1, r_2, \ldots, r_m)$ denote the $\lceil \frac{m}{2} \rceil$'th entry in the ordered list $r_{[1]} \leq r_{[2]} \leq \ldots \leq r_{[m]}$.

Consider a function $f\colon \mathbf{R}^d \to \mathbf{R} \cup \{+\infty\}$. The effective domain of f, denoted $\operatorname{dom} f$, consists of all points where f is finite. The function f is called μ-strongly convex if the perturbed function $f - \frac{\mu}{2}\|\cdot\|^2$ is convex. We say that f is L-smooth if it differentiable with L-Lipschitz continuous gradient. If f is both μ-strongly convex and L-smooth, then standard results in convex optimization (e.g., [37, § 2.1]) imply for all $x, y \in \mathbf{R}^d$ the bound

$$\langle \nabla f(y), x - y\rangle + \frac{\mu}{2}\|x - y\|^2 \leq f(x) - f(y) \leq \langle \nabla f(y), x - y\rangle + \frac{L}{2}\|x - y\|^2.$$

In particular, if y is the minimizer of f, denoted by $\bar{x}$, we have $\nabla f(\bar{x}) = 0$ and thus the two-sided bound:

$$\frac{\mu}{2}\|x - \bar{x}\|^2 \leq f(x) - f(\bar{x}) \leq \frac{L}{2}\|x - \bar{x}\|^2 \qquad \text{for all } x \in \mathbf{R}^d. \tag{2.1}$$

The ratio $\kappa := L/\mu$ is called the condition number of f.

Assumption 2.1. Throughout this work, we consider the optimization problem

$$\min_{x \in \mathbf{R}^d} \; f(x) \tag{2.2}$$

where the function $f\colon \mathbf{R}^d \to \mathbf{R} \cup \{+\infty\}$ is closed and μ-strongly convex. We denote the minimizer of f by $\bar{x}$ and its minimal value by $f^* := \min f$.

Let us suppose for the moment that the only access to f is by querying a black-box procedure that estimates $\bar{x}$. Namely following [20] we will call a procedure $\mathcal{D}(\varepsilon)$ a *weak distance oracle* for the problem (2.2) if it returns a point x satisfying

$$\mathbb{P}[\|x - \bar{x}\| \leq \varepsilon] \geq \frac{2}{3}. \tag{2.3}$$

We will moreover assume that when querying $\mathcal{D}(\varepsilon)$ multiple times, the returned vectors are all statistically independent. Weak distance oracles arise naturally in stochastic optimization both in streaming and offline settings. We will discuss specific examples in Sections 4 and 5. The numerical value 2/3 plays no real significance and can be replaced by any fraction greater than a half.

It is well known from [35, p. 243] and [20] that the low-confidence estimate (2.3) can be improved to a high confidence guarantee by a clustering technique. Following [20], we define the *robust distance estimator* $\mathcal{D}(\varepsilon, m)$ to be the following procedure (Algorithm 1).

Algorithm 1: Robust Distance Estimation $\mathcal{D}(\varepsilon, m)$

Input: access to a weak distance oracle $\mathcal{D}(\varepsilon)$ and trial count m.
Query m times the oracle $\mathcal{D}(\varepsilon)$ and let $Y = \{y_1, \ldots, y_m\}$ consist of the responses.
Step $i = 1, \ldots, m$:
  Compute $r_i = \min\{r \geq 0 : |B_r(y_i) \cap Y| > \frac{m}{2}\}$.
Set $i^* = \operatorname{argmin}_{i \in [1,m]} r_i$
Return y_{i^*}

Thus the robust distance estimator $\mathcal{D}(\varepsilon, m)$ first generates m statistically independent random points $y_1, \ldots, y_m$ by querying m times the weak distance oracle $\mathcal{D}(\varepsilon)$. Then the procedure computes the smallest radius ball around each point y_i that contains more than half of the generated points $\{y_1, \ldots, y_m\}$. Finally, the point y_{i^*} corresponding to the smallest such ball is returned. See Figure 1 for an illustration.

The intuition underlying the algorithm is that by Chernoff's bound, with high confidence, the ball $B_\varepsilon(\bar{x})$ will contain strictly more than $m/2$ of the generated points. Therefore in this event, the estimate $r_{i^*} < 2\varepsilon$ holds. Moreover since the two sets, $B_\varepsilon(\bar{x})$ and $B_{r_{i^*}}(y_{i^*})$ intersect, it follows that $\bar{x}$ and y_{i^*} are within a distance of 3ε of each other. For a complete argument, see [35, p. 243] or [20, Propositions 8,9].

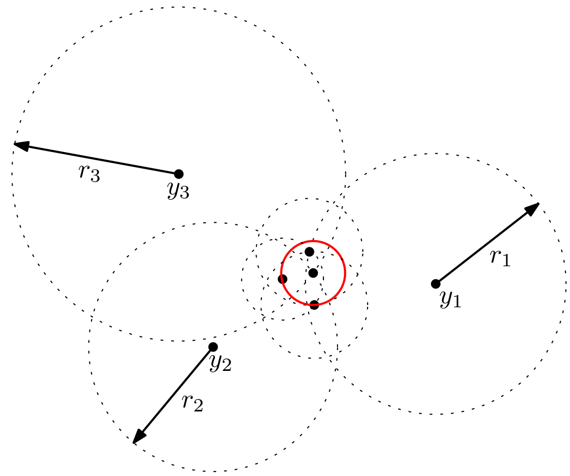

Figure 1: Illustration of the robust distance estimator $\mathcal{D}(\varepsilon, m)$.

Lemma 2.2 (Robust Distance Estimator)**.** *The point x returned by $\mathcal{D}(\varepsilon, m)$ satisfies*

$$\mathbb{P}\big(\|x - \bar{x}\| \leq 3\varepsilon\big) \geq 1 - \exp\left(-\frac{m}{18}\right).$$

We seek to understand how one may use a robust distance estimator $\mathcal{D}(\varepsilon, m)$ to compute a point x satisfying $f(x) - \min f \leq \delta$ with high probability, where $\delta > 0$ is a specified accuracy. As motivation, consider the case when f is also L-smooth. Then one immediate approach is to appeal to the upper bound in (2.1). Hence by Lemma 2.2, the point $x = \mathcal{D}(\varepsilon, m)$, with $\varepsilon = \sqrt{\frac{2\delta}{9L}}$, satisfies the guarantee

$$\mathbb{P}\left(f(x) - f^* \leq \delta\right) \;\geq\; \mathbb{P}\left(\|x - \bar{x}\| \leq 3\varepsilon\right) \;\geq\; 1 - \exp\left(-\frac{m}{18}\right).$$

We will follow an alternative approach, which can significantly decrease the overall cost in the regime $\kappa \gg 1$. The optimistic goal is to replace the accuracy $\varepsilon \approx \sqrt{\frac{\delta}{L}}$ used in the call to $\mathcal{D}(\varepsilon, m)$ by the potentially much larger quantity $\sqrt{\frac{\delta}{\mu}}$. The strategy we propose will apply a robust distance estimator $\mathcal{D}$ to a sequence of optimization problems that are better and better conditioned, thereby amortizing the overall cost. In the initial step, we will simply apply $\mathcal{D}$ to f with the low accuracy $\sqrt{\frac{\delta}{\mu}}$. In step i, we will apply $\mathcal{D}$ to a new function f^i, which has condition number $\kappa_i \approx \frac{L+\mu 2^i}{\mu+\mu 2^i}$, with accuracy $\varepsilon_i \approx \sqrt{\frac{\delta}{\mu+\mu 2^i}}$. Continuing this process for $T \approx \log_2\left(\frac{L}{\mu}\right)$ rounds, we arrive at accuracy $\varepsilon_T \approx \sqrt{\frac{\delta}{\mu+L}}$ and a function f^T that is nearly perfectly conditioned with $\kappa_T \leq 2$. In this way, the total cost is amortized over the sequence of optimization problems. The key of course is to control the error incurred by varying the optimization problems along the iterations.

3 Main result

The continuation procedure outlined at the end of the previous section can be succinctly described within the framework of an *inexact proximal point method*. Henceforth, fix an increasing sequence of penalties $\lambda_0, \ldots, \lambda_T$ and a sequence of centers $x_0, \ldots, x_T$. For each index $i = 0, \ldots, T$, define the quadratically perturbed functions and their minimizers:

$$f^i(x) := f(x) + \frac{\lambda_i}{2}\|x - x_i\|^2, \qquad \bar{x}_{i+1} := \operatorname*{argmin}_x f^i(x).$$

The exact proximal point method [30, 31, 40] proceeds by inductively declaring $x_i = \bar{x}_i$ for $i \geq 1$. Since computing $\bar{x}_i$ exactly is in general impossible, we will instead monitor the error $\|\bar{x}_i - x_i\|$. The following elementary result will form the basis for the rest of the paper. To simplify notation, we will set $\bar{x}_0 := \operatorname{argmin} f$ and $\lambda_{-1} := 0$, throughout.

Theorem 3.1 (Inexact proximal point method)**.** *For all $j \geq 0$, the following estimate holds:*

$$f^j(\bar{x}_{j+1}) - f^* \leq \sum_{i=0}^{j} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2. \tag{3.1}$$

Consequently, we have the error decomposition:

$$\boxed{f(x_{j+1}) - f^* \leq (f^j(x_{j+1}) - f^j(\bar{x}_{j+1})) + \sum_{i=0}^{j} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2.} \tag{3.2}$$

Moreover, if f is L-smooth, then for all $j \geq 0$ the estimate holds:

$$f(x_j) - f^* \leq \frac{L + \lambda_{j-1}}{2}\|\bar{x}_j - x_j\|^2 + \sum_{i=0}^{j-1} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2. \tag{3.3}$$

Proof. We first establish (3.1) by induction. For the base case $j = 0$, observe $\lambda_- = 0$ and

$$f^0(\bar{x}_1) = \min_x f^0(x) \leq f^0(\bar{x}_0) = f^* + \frac{\lambda_0}{2}\|\bar{x}_0 - x_0\|^2.$$

As the inductive assumption, suppose (3.1) holds up to iteration $j - 1$. We then conclude

$$\begin{aligned} f^j(\bar{x}_{j+1}) \leq f^j(\bar{x}_j) &= f(\bar{x}_j) + \frac{\lambda_j}{2}\|\bar{x}_j - x_j\|^2 \\ &\leq f^{j-1}(\bar{x}_j) + \frac{\lambda_j}{2}\|\bar{x}_j - x_j\|^2 \leq f^* + \sum_{i=0}^{j} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2, \end{aligned}$$

where the last inequality follows by the inductive assumption. This completes the proof of (3.1). To see (3.2), we observe using (3.1) the estimate

$$\begin{aligned} f(x_{j+1}) - f^* \leq f^j(x_{j+1}) - f^* &= (f^j(x_{j+1}) - f^j(\bar{x}_{j+1})) + f^j(\bar{x}_{j+1}) - f^* \\ &\leq (f^j(x_{j+1}) - f^j(\bar{x}_{j+1})) + \sum_{i=0}^{j} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2. \end{aligned}$$

Finally, if f is L-smooth, then f^j is $(L+\lambda_j)$-smooth. An analogous result to (2.1) yields

$$f^j(x_{j+1}) - f^j(\bar{x}_{j+1}) \le \frac{L+\lambda_j}{2}\|\bar{x}_{j+1} - x_{j+1}\|^2.$$

Inequality (3.3) follows from applying this bound in (3.2). $\square$

The main conclusion of Theorem 3.1 is the decomposition of the functional error described in (3.2). Namely, the estimate (3.2) upper bounds the error $f(x_{j+1}) - \min f$ as the sum of the suboptimality in the last step $f^T(x_{T+1}) - f^T(\bar{x}_{T+1})$ and the errors $\frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2$ incurred along the way. By choosing T sufficiently large, we can be sure that the function f^T is well-conditioned. Moreover in order to ensure that each term in the sum $\frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2$ is of order δ, it suffices to guarantee $\|\bar{x}_i - x_i\| \le \sqrt{\frac{2\delta}{\lambda_i}}$ for each index i. Since λ_i is an increasing sequence, it follows that we may gradually decrease the tolerance on the errors $\|\bar{x}_i - x_i\|$, all the while improving the conditioning of the functions we encounter. With this intuition in mind, we introduce the `proxBoost` procedure (Algorithm 2). The algorithm, and its latter modifications, depend on the amplitude sequence $\{\lambda_j\}_{j=1}^T$ governing the proximal regularization terms. To simplify notation, we will omit this sequence from the algorithm input and instead treat it as a global parameter specified in theorems.

Algorithm 2: `proxBoost`(δ, p, T)

Input: $\delta \ge 0$, $p \in (0,1)$, $T \in \mathbb{N}$
Set $\lambda_{-1} = 0$, $\varepsilon_{-1} = \sqrt{\frac{2\delta}{\mu}}$
Generate a point x_0 satisfying $\|x_0 - \bar{x}_0\| \le \varepsilon_{-1}$ with probability $1-p$.
for $j = 0, \ldots, T-1$ **do**
  Set $\varepsilon_j = \sqrt{\frac{2\delta}{\mu+\lambda_j}}$
  Generate a point x_{j+1} satisfying

$$\mathbb{P}\left[\|x_{j+1} - \bar{x}_{j+1}\| \le \varepsilon_j \mid E_j\right] \ge 1-p, \tag{3.4}$$

  where E_j denotes the event $E_j := \left\{x_i \in B_{\varepsilon_{i-1}}(\bar{x}_i) \text{ for all } i \in [0,j]\right\}$.
end
Generate a point x_{T+1} satisfying

$$\mathbb{P}\left[f^T(x_{T+1}) - \min f^T \le \delta \mid E_T\right] \ge 1-p. \tag{3.5}$$

Return x_{T+1}

Thus `proxBoost` consists of three stages, which we now examine in detail.

Stage I: Initialization. Algorithm 2 begins by generating a point x_0 that is a distance of $\sqrt{\frac{2\delta}{\mu}}$ away from the minimizer of f with probability $1-p$. This task can be achieved by applying a robust distance estimator on f, as discussed in Section 2.

Stage II: Proximal iterations. In each subsequent iteration, x_{j+1} is defined to be a point that is within a radius of $\varepsilon_j = \sqrt{\frac{2\delta}{\mu+\lambda_j}}$ from the minimizer of f^j with probability $1-p$ conditioned on the event E_j. The event E_j encodes that each previous iteration was successful in the sense that the point x_i indeed lies inside the ball $B_{\varepsilon_{i-1}}(\bar{x}_i)$ for all $i = 0, \ldots, j$. Thus x_{j+1} can be determined by a procedure that conditioned on the event E_j is a robust distance estimator on the function f^j.

Stage III: Cleanup. In the final step, the algorithm outputs a δ-minimizer of f^T with probability $1-p$ conditioned on the event E_T. In particular, if f is L-smooth then we may use a robust distance estimator on f^T directly. Namely, taking into account the upper bound in (2.1), we may declare x_{T+1} to be any point satisfying

$$\mathbb{P}\left[\|x_{T+1} - \bar{x}_{T+1}\| \leq \sqrt{\tfrac{2\delta}{L+\lambda_T}} \mid E_T\right] \geq 1-p.$$

Notice that by choosing λ_T sufficiently large, we may ensure that the condition number $\frac{\mu+\lambda_T}{L+\lambda_T}$ of f^T is arbitrarily close to one. If f is not smooth, such as when constraints or additional regularizers are present, we can not use a robust distance estimator in the cleanup stage. We will see in Section 6 a different approach for convex composite problems, based on a modified robust distance estimation technique.

The following theorem summarizes the guarantees of the `proxBoost` procedure.

Theorem 3.2 (Proximal Boost). *Fix a constant $\delta > 0$, a probability of failure $p \in (0,1)$ and a natural number $T \in \mathbb{N}$. Then with probability at least $1-(T+2)p$, the point $x_{T+1} =$ `proxBoost`(δ, p, T) satisfies*

$$f(x_{T+1}) - \min f \leq \delta\left(1 + \sum_{i=0}^{T} \frac{\lambda_i}{\mu + \lambda_{i-1}}\right). \tag{3.6}$$

Proof. We first prove by induction the estimate

$$\mathbb{P}[E_t] \geq 1-(t+1)p \qquad \text{for all } t = 0, \ldots, T. \tag{3.7}$$

The base case $t = 0$ is immediate from the definition of x_0. Suppose now that (3.7) holds for some index $t-1$. Then the inductive assumption and the definition of x_t yield

$$\mathbb{P}[E_t] = \mathbb{P}[E_t \big| E_{t-1}]\mathbb{P}[E_{t-1}] \geq (1-p)(1-tp) \geq 1-(t+1)p,$$

thereby completing the induction. Thus the inequalities (3.7) hold. Define the event

$$F = \{f^T(x_{T+1}) - \min f^T \leq \delta\}.$$

We therefore deduce

$$\mathbb{P}[F \cap E_T] = \mathbb{P}[F \mid E_T] \cdot \mathbb{P}[E_T] \geq (1-(T+1)p)(1-p) \geq 1-(T+2)p.$$

Suppose now that the event $F \cap E_T$ occurs. Then using the estimate (3.2), we conclude

$$f(x_{T+1}) - \min f \leq (f^T(x_{T+1}) - f^T(\bar{x}_{T+1})) + \sum_{i=0}^{T} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2 \leq \delta + \sum_{i=0}^{T} \frac{\delta\lambda_i}{\mu + \lambda_{i-1}},$$

where the last inequality uses the definitions of x_{T+1} and ε_j. This completes the proof. $\square$

Looking at the estimate (3.6), we see that the final error $f(x_{T+1}) - \min f$ is controlled by the sum $\sum_{i=0}^{T} \frac{\lambda_i}{\mu+\lambda_{i-1}}$. A moment of thought yields an appealing choice $\lambda_i = \mu 2^i$ for the proximal parameters. Indeed, then every element in the sum $\frac{\lambda_i}{\mu+\lambda_{i-1}}$ is upper bounded by two. Moreover, if f is L-smooth, then the condition number $\frac{L+\lambda_T}{\mu+\lambda_T}$ of f^T is upper bounded by two after only $T = \lceil \log(L/\mu) \rceil$ rounds.

Corollary 3.3 (Proximal boost with geometric decay)**.** *Fix an iteration count T, a target accuracy $\epsilon > 0$, and a probability of failure $p \in (0,1)$. Define the algorithm parameters:*

$$\delta = \frac{\epsilon}{2+2T} \qquad \textit{and} \qquad \lambda_i = \mu 2^i \qquad \forall i \in [0, T].$$

Then the point $x_{T+1} = \texttt{proxBoost}(\delta, p, T)$ satisfies

$$\mathbb{P}(f(x_{T+1}) - \min f \leq \epsilon) \geq 1 - (T+2)p.$$

In the next two sections, we seed the `proxBoost` procedure with (accelerated) stochastic gradient algorithms and methods based on empirical risk minimization. The reader, however, should keep in mind that `proxBoost` is entirely agnostic to the inner workings of the robust distance estimators it uses. The only point to be careful about is that some distance estimators (e.g., when using stochastic gradient methods) require auxiliary quantities as input, such as an upper estimate on the function gap at the initial point. Therefore, we may have to update such estimates along the iterations of `proxBoost`.

4 Consequences for empirical risk minimization

In this section, we explore the consequences of the `proxBoost` algorithm for empirical risk minimization. Setting the stage, fix a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ and equip $\mathbf{R}^d$ with the Borel σ-algebra. Consider the optimization problem

$$\min_x f(x) = \mathbb{E}_{z\sim\mathcal{P}}\left[f(x,z)\right], \tag{4.1}$$

where $f\colon \mathbf{R}^d \times \Omega \to \mathbf{R}_+$ is a measurable nonnegative function. A common approach to problems of the form (4.1) is based on empirical risk minimization. Namely, one collects i.i.d. samples $z_1, \ldots, z_n \sim \mathcal{P}$ and minimizes the empirical average

$$\min_x f_S(x) := \frac{1}{n}\sum_{i=1}^{n} f(x, z_i). \tag{4.2}$$

A central question is to determine the number n of samples that would ensure that the minimizer x_S of the empirical risk has low generalization error $f(x_S) - \min f$, with reasonably high probability. There is a vast literature on this subject; some representative works include [5,20,41,42]. We build here on the work of Hsu-Sabato [20], who specifically focused on high confidence guarantees for smooth strongly convex minimization. As in the previous sections, we let $\bar{x}$ be a minimizer of f and define the shorthand $f^* = \min f$.

Assumption 4.1. Following [20], we make the following assumptions on the loss.

1. **(Strong convexity)** There exist a real $\mu > 0$ and a natural number $N \in \mathbb{N}$ such that:
   (a) the population loss f is μ-strongly convex,
   (b) the empirical loss $x \mapsto f_S(x)$ is μ-strongly convex with probability at least 5/6, whenever $|S| \geq N$.
2. **(Smoothness)** There exist constants $L, \hat{L} > 0$ such that:
   (a) for a.e. $z \sim \mathcal{P}$, the loss $x \mapsto f(x, z)$ is $\hat{L}$-smooth,
   (b) the population objective $x \mapsto f(x)$ is L-smooth. (It holds that $L \leq \hat{L}$.)

In addition, we assume $f^* := \min f > 0$.

The following result proved in [20, Theorem 15] shows that the empirical risk minimizer is a weak distance oracle for the problem (4.1).

Lemma 4.2. *Fix an i.i.d. sample $z_1, \ldots, z_n \sim \mathcal{P}$ of size $n \geq N$. Suppose Assumption 4.1 holds. Then the minimizer x_S of the empirical risk* (4.2) *satisfies the bound:*

$$\mathbb{P}\left[\|x_S - \bar{x}\| \leq \sqrt{\frac{96\hat{L}f^*}{n\mu^2}}\right] \geq 2/3.$$

In particular, using Algorithm 1 one may turn empirical risk minimization into a robust distance estimator for the problem (4) using a total of mn samples. Let us estimate the function value at the generated point by a direct application of smoothness. Appealing to Lemma 2.2 and the two-sided bound (2.1), we deduce that with probability $1 - \exp(-m/18)$ the procedure will return a point x satisfying

$$f(x) \leq \left(1 + \frac{432\hat{L}L}{n\mu^2}\right) f^*.$$

Observe that this is an estimate of *relative error*. In particular, let $p \in (0,1)$ be some acceptable probability of failure and let $\gamma > 0$ be a desired level of relative accuracy. Then setting $m = \lceil 18\ln(1/p)\rceil$ and $n \geq \max\{\frac{432\hat{\kappa}\kappa}{\gamma}, N\}$, we conclude that x satisfies

$$\mathbb{P}[f(x) \leq (1+\gamma)f^*] \geq 1 - p, \tag{4.3}$$

while the overall sample complexity is

$$\left\lceil 18 \ln \left(\frac{1}{p} \right) \right\rceil \cdot \max \left\{ \left\lceil \frac{432 \hat{\kappa} \kappa}{\gamma} \right\rceil, N \right\}, \tag{4.4}$$

where $\hat{\kappa} = \hat{L}/\mu$ and $\kappa = L/\mu$. This is exactly the result [20, Corollary 16].

We will now see how to find a point x satisfying (4.3) with significantly fewer samples by embedding empirical risk minimization within `proxBoost`. Algorithm 3 encodes the empirical risk minimization process on a quadratically regularized problem. Algorithm 4 is the robust distance estimator induced by Algorithm 3. Finally, Algorithm 5 is the `proxBoost` algorithm specialized to empirical risk minimization.

Algorithm 3: `ERM`(n, λ, x)

Input: sample count $n \in \mathbb{N}$, center $x \in \mathbf{R}^d$, amplitude $\lambda > 0$.
Generate i.i.d. samples $z_1, \ldots, z_n \sim \mathcal{P}$ and compute the minimizer $\bar{y}$ of

$$\min_y \; \frac{1}{n} \sum_{i=1}^{n} f(y, z_i) + \frac{\lambda}{2} \|y - x\|^2.$$

Return $\bar{y}$

Algorithm 4: `ERM-R`(n, m, λ, x)

Input: sample count $n \in \mathbb{N}$, trial count $m \in \mathbb{N}$, center $x \in \mathbf{R}^d$, amplitude $\lambda > 0$.
Query m times `ERM`(n, λ, x) and let $Y = \{y_1, \ldots, y_m\}$ consist of the responses.
Step $j = 1, \ldots, m$:
    Compute $r_i = \min\{r \geq 0 : |B_r(y_i) \cap Y| > \frac{m}{2}\}$.
Set $i^* = \mathrm{argmin}_{i \in [1,m]} r_i$
Return y_{i^*}

Algorithm 5: `BoostERM`(γ, T, m)

Input: $T, m \in \mathbb{N}$, $\gamma > 0$
Set $\lambda_{-1} = 0$, $x_{-1} = 0$, $n_{-1} = \frac{432 \hat{L}}{\gamma \mu}$
Step $j = 0, \ldots, T$:
    $x_j =$ `ERM-R`$(n_{j-1}, m, \lambda_{j-1}, x_{j-1})$
    $n_j = 432 \left\lceil \frac{\hat{L} + \lambda_j}{\mu + \lambda_j} \left(\frac{1}{\gamma} + \sum_{i=0}^{j} \frac{\lambda_i}{\mu + \lambda_{i-1}} \right) \right\rceil \vee N$
Return $x_{T+1} =$ `ERM-R`$(\frac{L + \lambda_T}{\mu + \lambda_T} \cdot n_T, m, \lambda_T, x_T)$

Using Theorem 3.2, we can now prove the following result.

Theorem 4.3 (Efficiency of `BoostERM`)**.** *Fix a target relative accuracy $\gamma > 0$ and numbers $T, m \in \mathbb{N}$. Then with probability at least $1 - (T+2) \exp\left(-\frac{m}{18}\right)$, the point $x_{T+1} =$*

`BoostERM`(γ, T, m) *satisfies*

$$f(x_{T+1}) - f^* \le \left(1 + \sum_{i=0}^{T} \frac{\lambda_i}{\mu + \lambda_{i-1}}\right) \gamma f^*.$$

Proof. We will verify that Algorithm 5 is an instantiation of Algorithm 2 with $\delta = \gamma f^*$ and $p = \exp(-\frac{m}{18})$. More precisely, we will prove by induction that with this choice of p and δ, the iterates x_j satisfy (3.4) for each index $j = 0, \ldots, T$ and x_{T+1} satisfies (3.5). As the base case, consider the evaluation $x_0 = \texttt{ERM-R}(n_{-1}, m, \lambda_{-1}, x_{-1})$ where x_{-1} can be arbitrary since $\lambda_{-1} = 0$. Then Lemma 2.2 and Theorem 4.2 guarantee

$$\mathbb{P}\left[\|x_0 - \bar{x}_0\| \le 3\sqrt{\frac{96\hat{L}f^*}{n_{-1}\mu^2}}\right] \ge 1 - \exp\left(-\frac{m}{18}\right).$$

Taking into account the definitions of n_{-1} in Algorithm 5 and ε_{-1} in Algorithm 2, we deduce

$$\mathbb{P}\left[\|x_0 - \bar{x}_0\| \le \epsilon_{-1}\right] \ge 1 - p,$$

as claimed. As an inductive hypothesis, suppose that (3.4) holds for $x_0, x_1, \ldots, x_{j-1}$. We will prove it holds for $x_j = \texttt{ERM-R}(n_{j-1}, m, \lambda_{j-1}, x_{j-1})$. To this end, suppose that the event E_{j-1} occurs. Then by the same reasoning as in the base case, the point x_j satisfies

$$\mathbb{P}\left[\|x_j - \bar{x}_j\| \le 3\sqrt{\frac{96(\hat{L} + \lambda_{j-1})f^{j-1}(\bar{x}_j)}{n_{j-1}(\mu + \lambda_{j-1})^2}}\right] \ge 1 - \exp\left(-\frac{m}{18}\right). \tag{4.5}$$

Now, using (3.1) and the inductive assumption that $\|x_i - \bar{x}_i\| \le \varepsilon_{i-1} = \sqrt{\frac{2\delta}{\mu + \lambda_{i-1}}}$ for all $i \in [0, j-1]$ (conditioned on E_{j-1}), we have

$$f^{j-1}(\bar{x}_j) - f^* \le \sum_{i=0}^{j-1} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2 \le \delta \sum_{i=0}^{j-1} \frac{\lambda_i}{\mu + \lambda_{i-1}},$$

which, together with $\delta = \gamma f^*$, implies

$$f^{j-1}(\bar{x}_j) \le f^* + \delta \sum_{i=0}^{j-1} \frac{\lambda_i}{\mu + \lambda_{i-1}} = \left(1 + \gamma \sum_{i=0}^{j-1} \frac{\lambda_i}{\mu + \lambda_{i-1}}\right) f^*.$$

Combining this inequality with (4.5), we conclude that conditioned on the event E_{j-1}, we have with probability $1 - p$ the guarantee

$$\frac{\mu + \lambda_{j-1}}{2}\|x_j - \bar{x}_j\|^2 \le \frac{432(\hat{L} + \lambda_{j-1})(1 + \gamma \sum_{i=0}^{j-1} \frac{\lambda_i}{\mu + \lambda_{i-1}})}{n_{j-1}(\mu + \lambda_{j-1})} \cdot f^* \le \gamma f^* = \delta, \tag{4.6}$$

where the last inequality follows from the definition of n_{j-1}. This implies that the estimate (3.4) holds for x_j with $\epsilon_{j-1} = \sqrt{\frac{2\delta}{\mu + \lambda_{j-1}}}$. Therefore, it holds for all iterates $x_0, \ldots, x_T$, as needed. Suppose now that that event E_T occurs. Then by exactly the same reasoning that led to (4.6), and considering the extra factor $\frac{L+\lambda_T}{\mu+\lambda_T}$ multiplied to n_T in the last call of ERM-R, we have the estimate

$$\frac{\mu+\lambda_T}{2}\|x_{T+1}-\bar{x}_{T+1}\|^2 \leq \frac{\mu+\lambda_T}{L+\lambda_T}\gamma f^*.$$

Using smoothness, we therefore deduce $f^T(x_{T+1}) - \min f^T \leq \gamma f^* = \delta$, as claimed. An application of Theorem 3.2 completes the proof. □

Finally, using the proximal parameters $\lambda_i = \mu 2^i$ yields the following guarantee.

Corollary 4.4 (Efficiency of BoostERM with geometric decay)**.** *Fix a target relative accuracy $\gamma' > 0$ and a probability of failure $p \in (0,1)$. Define the algorithm parameters:*

$$T = \lceil \log_2(\kappa) \rceil, \qquad m = \left\lceil 18 \ln\left(\frac{T+2}{p}\right)\right\rceil, \qquad \gamma = \frac{\gamma'}{2+2T}, \qquad \lambda_i = \mu 2^i.$$

Then with probability of at least $1-p$, the point $x_{T+1} =$ BoostERM(γ, T, m) satisfies $f(x^{T+1}) \leq (1+\gamma')f^*$. Moreover, the total number of samples used by the algorithm is

$$\mathcal{O}\left(\ln(\kappa)\ln\left(\frac{\ln(\kappa)}{p}\right)\cdot \max\left\{\left(1+\tfrac{1}{\gamma'}\right)\hat{\kappa}\ln(\kappa), N\right\}\right).$$

Notice that the sample complexity provided by Corollary 4.4 is an order of magnitude better than (4.4) in terms of the dependence on the condition numbers $\hat{\kappa}$ and κ.

5 Consequences for stochastic approximation

We next investigate the consequences of `proxBoost` for stochastic approximation. Namely, we will seed `proxBoost` with the robust distance estimator, induced by the stochastic gradient method and its accelerated variant. An important point is that the sample complexity of stochastic gradient methods depends on the initialization quality $f(x_0) - f^*$. Consequently, in order to know how many iterations are needed to reach a desired accuracy $\mathbb{E}[f(x_i)] - f^* \leq \delta$, we must have available an upper bound on the initialization quality $\Delta \geq f(x_0) - f^*$. Therefore, we will have to dynamically update an estimate of the initialization quality for each proximal subproblem along the iterations of `proxBoost`. The following assumption formalizes this idea.

Assumption 5.1. Consider the proximal minimization problem

$$\min_y \ \varphi_x(y) := f(y) + \frac{\lambda}{2}\|y-x\|^2,$$

Let $\Delta > 0$ be a real number satisfying $\varphi_x(x) - \min \varphi_x \leq \Delta$. We will let $\texttt{Alg}(\delta, \lambda, \Delta, x)$ be a procedure that returns a point y satisfying

$$\mathbb{P}[\varphi_x(y) - \min \varphi_x \leq \delta] \geq \frac{2}{3}.$$

Clearly, $\texttt{Alg}(\delta, \lambda, \Delta, x)$ is a minimization oracle in the sense of (1.3). Since the proximal function φ_x is $(\mu+\lambda)$-strongly convex, it has a unique minimizer $\bar{y}_x$ and satisfies

$$\frac{\mu+\lambda}{2}\|y-\bar{y}_x\|^2 \leq \varphi_x(y) - \min \varphi_x.$$

Therefore, $\texttt{Alg}(\delta, \lambda, \Delta, x)$ is a weak distance oracle, in the sense that $\mathbb{P}(\|y-\bar{y}_x\| \leq \varepsilon) \geq \frac{2}{3}$ with $\varepsilon = \sqrt{\frac{2\delta}{\mu+\lambda}}$. Following the procedure in Section 2, we may turn it into a robust distance estimator for minimizing φ_x, as long as Δ upper bounds the initialization error. We record the robust distance estimator induced by $\texttt{Alg}(\cdot)$ as Algorithm 6.

Algorithm 6: $\texttt{Alg-R}(\delta, \lambda, \Delta, x, m)$

Input: accuracy $\delta > 0$, amplitude $\lambda > 0$, upper bound $\Delta > 0$, center $x \in \mathbf{R}^d$, trial count $m \in \mathbb{N}$.
Query m times $\texttt{Alg}(\delta, \lambda, \Delta, x)$ and let $Y = \{y_1, \ldots, y_m\}$ consist of the responses.
Step $j = 1, \ldots, m$:
  Compute $r_i = \min\{r \geq 0 : |B_r(y_i) \cap Y| > \frac{m}{2}\}$.
Set $i^* = \operatorname{argmin}_{i\in[1,m]} r_i$
Return y_{i^*}

Henceforth, in addition to Assumptions 2.1 and 5.1, we assume that f is L-smooth and set $\kappa = \frac{L}{\mu}$. It is then straightforward to instantiate $\texttt{proxBoost}$ with the robust distance estimator $\texttt{Alg-R}$. We record the resulting procedure as Algorithm 7.

Algorithm 7: $\texttt{BoostAlg}(\delta, \Delta_{\text{in}}, x_{\text{in}}, T, m)$

Input: accuracy $\delta > 0$, upper bound $\Delta_{\text{in}} > 0$, center $x_{\text{in}} \in \mathbf{R}^d$, and $m, T \in \mathbb{N}$
Set $\lambda_{-1} = 0$, $\Delta_{-1} = \Delta_{\text{in}}$, $x_{-1} = x_{\text{in}}$
Step $j = 0, \ldots, T$:
  $x_j = \texttt{Alg-R}(\delta/9, \lambda_{j-1}, \Delta_{j-1}, x_{j-1}, m)$
  $\Delta_j = \delta\left(\frac{L+\lambda_{j-1}}{\mu+\lambda_{j-1}} + \sum_{i=0}^{j-1}\frac{\lambda_i}{\mu+\lambda_{i-1}}\right)$
Return $x_{T+1} = \texttt{Alg-R}(\frac{\mu+\lambda_T}{L+\lambda_T}\cdot\frac{\delta}{9}, \lambda_T, \Delta_T, x_T, m)$

We can now prove the following theorem on the efficiency of Algorithm 7. The proof is almost a direct application of Theorem 3.2. The only technical point is to verify that for all indices j, the quantity Δ_j is a valid upper bound on the initialization error $f^j(x_j) - \min f^j$ in the event E_j (defined in Algorithm 2).

Theorem 5.2 (Efficiency of $\texttt{BoostAlg}$)**.** *Fix an arbitrary point $x_{\text{in}} \in \mathbf{R}^d$ and let Δ_{in} be any constant satisfying $\Delta_{\text{in}} \geq f(x_{\text{in}}) - \min f$. Fix natural numbers $T, m \in \mathbb{N}$. Then with probability at least $1-(T+2)\exp\left(-\frac{m}{18}\right)$, the point $x_{T+1} = \texttt{BoostAlg}(\delta, \Delta_{\text{in}}, x_{\text{in}}, T, m)$ satisfies*

$$f(x_{T+1}) - \min f \leq \delta\left(1 + \sum_{i=0}^{T}\frac{\lambda_i}{\mu+\lambda_{i-1}}\right).$$

Proof. We will verify that Algorithm 7 is an instantiation of Algorithm 2 with $p = \exp(-\frac{m}{18})$. More precisely, we will prove by induction that with this choice of p, the iterates x_j satisfy (3.4) for each index $j = 0, \ldots, T$ and x_{T+1} satisfies (3.5). For the base case $j = 0$, Lemma 2.2 guarantees that with probability $1 - p$, the point x_0 produced by the robust distance estimator `Alg-R` satisfies

$$\|x_0 - \bar{x}_0\| \leq 3\sqrt{\frac{2 \cdot \delta/9}{\mu}} = \varepsilon_{-1}.$$

As an inductive hypothesis, suppose that (3.4) holds for the iterates $x_0, \ldots, x_{j-1}$ for some $j \geq 1$. We will prove it holds for x_j. To this end, suppose that the event E_{j-1} occurs. Then using (3.3) we deduce

$$\begin{aligned} f(x_{j-1}) - f^* &\leq \frac{L + \lambda_{j-2}}{2}\|\bar{x}_{j-1} - x_{j-1}\|^2 + \sum_{i=0}^{j-2} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2 \\ &\leq \frac{\delta(L + \lambda_{j-2})}{\mu + \lambda_{j-2}} + \sum_{i=0}^{j-2} \frac{\delta\lambda_i}{\mu + \lambda_{i-1}} = \Delta_{j-1}, \end{aligned}$$

where the second inequality follows from $x_i \in B_{\varepsilon_{i-1}}(\bar{x}_i)$ with $\varepsilon_{i-1} = \sqrt{\frac{2\delta}{\mu+\lambda_{i-1}}}$ for all $i \in [0, j-1]$. By examining the definition of f^{j-1}, we deduce $f^{j-1}(x_{j-1}) = f(x_{j-1})$ and $\min f^{j-1} \geq \min f = f^*$, which imply

$$f^{j-1}(x_{j-1}) - \min f^{j-1} \leq f(x_{j-1}) - f^* \leq \Delta_{j-1}. \tag{5.1}$$

That is, Δ_{j-1} is an upper bound on the initial gap $f^{j-1}(x_{j-1}) - \min f^{j-1}$ for all j whenever the event E_{j-1} occurs. Moreover Lemma 2.2 guarantees that conditioned on E_{j-1} with probability $1 - p$, the following estimate holds:

$$\|x_j - \bar{x}_j\| \leq 3\sqrt{\frac{2 \cdot \delta/9}{\mu + \lambda_{j-1}}} = \varepsilon_{j-1}.$$

Thus the condition (3.4) holds for the iterate x_j, as desired.

Now suppose that the event E_T holds. Then exactly the same reasoning that led to (5.1) yields the guarantee $f^T(x_T) - \min f^T \leq \Delta_T$. Therefore Lemma 2.2 guarantees that with probability $1 - p$ conditioned on E_T, we have

$$\|x_{T+1} - \bar{x}_{T+1}\| \leq 3\sqrt{\frac{2}{\mu + \lambda_T} \cdot \frac{\delta}{9} \cdot \frac{\mu + \lambda_T}{L + \lambda_T}} = \sqrt{\frac{2\delta}{L + \lambda_T}}.$$

Taking into account the fact that f^T is $(L + \lambda_T)$-smooth, we therefore deduce

$$\mathbb{P}[f^T(x_{T+1}) - \min f^T \leq \delta \mid E_T] \geq 1 - p,$$

thereby establishing (3.5). An application of Theorem 3.2 completes the proof. $\square$

When using the proximal parameters $\lambda_i = \mu 2^i$, we obtain the following guarantee.

Corollary 5.3 (Efficiency of `BoostAlg` with geometric decay)**.** *Fix an arbitrary point $x_{\text{in}} \in \mathbf{R}^d$ and let Δ_{in} be any upper bound $\Delta_{\text{in}} \geq f(x_{\text{in}}) - \min f$. Fix a target accuracy $\epsilon > 0$ and probability of failure $p \in (0,1)$, and set the algorithm parameters*

$$T = \lceil \log_2(\kappa) \rceil, \qquad m = \left\lceil 18 \ln\left(\frac{2+T}{p}\right)\right\rceil, \qquad \delta = \frac{\epsilon}{2+2T}, \qquad \lambda_i = \mu 2^i.$$

Then the point $x_{T+1} = \texttt{BoostAlg}(\delta, \Delta_{\text{in}}, x_{\text{in}}, T, m)$ satisfies

$$\mathbb{P}(f(x_{T+1}) - \min f \leq \epsilon) \geq 1 - p.$$

Moreover, the total number of calls to $\texttt{Alg}(\cdot)$ is

$$\left\lceil 18 \ln\left(\frac{\lceil 2 + \log_2(\kappa)\rceil}{p}\right)\right\rceil \lceil 2 + \log_2(\kappa)\rceil,$$

while the initialization errors satisfy

$$\max_{i=0,\ldots,T+1} \Delta_i \leq \frac{\kappa + 1 + 2\lceil \log_2(\kappa)\rceil}{2 + 2\lceil \log_2(\kappa)\rceil}\epsilon.$$

We now concretely describe how to use (accelerated) stochastic gradient methods as `Alg(·)` within `proxBoost`.

Illustration: robust (accelerated) stochastic gradient methods

Following the standard literature on streaming algorithms, we suppose that the only access to f is through a stochastic gradient oracle. Namely, fix a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ and let $G\colon \mathbf{R}^d \times \Omega \to \mathbf{R}$ be a measurable map satisfying

$$\mathbb{E}_z G(x,z) = \nabla f(x) \qquad \text{and} \qquad \mathbb{E}_z \|G(x,z) - \nabla f(x)\|^2 \leq \sigma^2.$$

We suppose that for any point x, we may sample $z \in \Omega$ and compute the vector $G(x,z)$, which serves as an unbiased estimator of the gradient $\nabla f(x)$. The performance of standard numerical methods within this model of computation is judged by their sample complexity—the number of stochastic gradient evaluations $G(x,z)$ with $z \sim \mathcal{P}$ required by the algorithm to produce an approximate minimizer of the problem.

Fix an initial point x_{in} and let $\Delta_{\text{in}} > 0$ satisfy $\Delta_{\text{in}} \geq f(x_0) - f^*$. It is well known that an appropriately modified stochastic gradient method can generate a point x satisfying $\mathbb{E}f(x) - f^* \leq \epsilon$ with sample complexity

$$\mathcal{O}\left(\kappa \log\left(\frac{\Delta_{\text{in}}}{\epsilon}\right) + \frac{\sigma^2}{\mu\epsilon}\right). \tag{5.2}$$

The accelerated stochastic gradient method of [18, Multi-stage AC-SA, Proposition 6] and the simplified optimal algorithm of [25, Restarted Algorithm C, Corollary 9] have the substantially better sample complexity

$$\mathcal{O}\left(\sqrt{\kappa}\log\left(\frac{\Delta_{\rm in}}{\epsilon}\right)+\frac{\sigma^2}{\mu\epsilon}\right). \tag{5.3}$$

Clearly, we may use either of these two procedures as $\texttt{Alg}(\cdot)$ within the `proxBoost` framework. Indeed, using Corollary 5.3, we deduce that the two resulting algorithms will find a point x satisfying

$$\mathbb{P}[f(x)-f^*\leq\epsilon]\geq 1-p$$

with sample complexities

$$\mathcal{O}\left(\ln(\kappa)\ln\left(\frac{\ln\kappa}{p}\right)\cdot\left(\kappa\ln\left(\frac{\Delta_{\rm in}\ln(\kappa)}{\epsilon}\vee\kappa\right)+\frac{\sigma^2\ln(\kappa)}{\mu\epsilon}\right)\right), \tag{5.4}$$

and

$$\mathcal{O}\left(\ln(\kappa)\ln\left(\frac{\ln\kappa}{p}\right)\cdot\left(\sqrt{\kappa}\ln\left(\frac{\Delta_{\rm in}\ln(\kappa)}{\epsilon}\vee\kappa\right)+\frac{\sigma^2\ln(\kappa)}{\mu\epsilon}\right)\right), \tag{5.5}$$

for the unaccelerated and accelerated methods, respectively. Thus, `proxBoost` endows the stochastic gradient method and its accelerated variant with high confidence guarantees at an overhead cost that is only polylogarithmic in κ and logarithmic in $1/p$.

6 Extension to convex composite problems

One limitation of the techniques presented in Sections 4 and 5 is that the function f to be minimized was assumed to be smooth. In particular, these techniques can not accommodate constraints or nonsmooth regularizers. To illustrate the difficulty, consider the task of minimizing a smooth and strongly convex function f over a closed convex set $\mathcal{X}$. The current approach heavily relies on the two-sided bound (2.1), which guarantees that the function gap $f(x)-f^*$ and the squared distance to the solution $\|x-\bar{x}\|^2$ are proportional up to multiplication by the condition number of f. When a constraint set $\mathcal{X}$ is present, the left inequality of (2.1) still holds, but the right inequality is typically false. In particular, in the clean up stage of `proxBoost`, it is unclear how to turn low probability guarantees on the function gap to high probability guarantees using robust distance estimation. In this section, we show how to overcome this difficulty and extend the aforementioned techniques to convex composite optimization problems.

6.1 Geometric intuition in the constrained case

Before delving into the details, it is instructive to first focus on the constrained setting, where no additional regularizers are present. This is the content of this section. In section 6.2, we formally describe the algorithm for regularized convex optimization problems in full generality and prove correctness. Consequently, the reader may safely skip to Section 6.2, without losing continuity.

Setting the stage, consider the optimization problem

$$\min_x \; g(x) \quad \text{subject to} \quad x \in \mathcal{X},$$

where $g\colon \mathbf{R}^d \to \mathbf{R}$ is μ-strongly convex and L-smooth and $\mathcal{X}$ is a closed convex set. In line with the previous sections, let $\bar{x}$ be the minimizer of the problem and let $\kappa = \frac{L}{\mu}$ denote the condition number. Suppose that we have available an algorithm $\mathcal{M}(\epsilon)$ that generates a point x_ϵ satisfying

$$\mathbb{P}\left(g(x_\epsilon) - \min_{x \in \mathcal{X}} g \leq \epsilon\right) \geq \frac{2}{3}. \tag{6.1}$$

Our immediate goal is to explain how to efficiently boost this low-probability guarantee to a high confidence outcome, albeit with the degraded accuracy $\kappa\epsilon$.

We begin as in the unconstrained setting with the two-sided bound:

$$\langle \nabla g(\bar{x}), x - \bar{x}\rangle + \frac{\mu}{2}\|x - \bar{x}\|^2 \leq g(x) - g(\bar{x}) \leq \langle \nabla g(\bar{x}), x - \bar{x}\rangle + \frac{L}{2}\|x - \bar{x}\|^2 \tag{6.2}$$

for all $x \in \mathcal{X}$. In particular, if the minimizer $\bar{x}$ lies in the interior of $\mathcal{X}$ the gradient $\nabla g(\bar{x})$ vanishes and the estimate (6.2) reduces to (2.1). In the more general constrained setting, however, the additive term $\langle \nabla g(\bar{x}), x - \bar{x}\rangle$ plays an important role. Note that optimality conditions at $\bar{x}$ immediately imply that this term is nonnegative

$$\langle \nabla g(\bar{x}), x - \bar{x}\rangle \geq 0 \qquad \text{for all } x \in \mathcal{X}.$$

Moreover, we see from the estimate (6.2) that the point x_ϵ returned by $\mathcal{M}(\epsilon)$, with probability $2/3$, lies in the region

$$\Lambda := \left\{x \in \mathcal{X} : \|x - \bar{x}\| \leq \sqrt{\frac{2\epsilon}{\mu}} \quad \text{and} \quad 0 \leq \langle \nabla g(\bar{x}), x - \bar{x}\rangle \leq \epsilon\right\}. \tag{6.3}$$

Thus x_ϵ simultaneously lies in the ball around $\bar{x}$ of radius $\sqrt{2\epsilon/\mu}$ and is sandwiched between two parallel hyperplanes with normal $\nabla g(\bar{x})$. See Figure 2 for an illustation.

Naturally, our goal is to generate a point x that lies in Λ, or a slight perturbation thereof, with probability $1 - p$. As the first attempt, suppose for the moment that we know the value of the gradient $\nabla g(\bar{x})$. Then we can define the metric

$$\rho(x, x') = \max\left\{\sqrt{\frac{\epsilon\mu}{2}} \cdot \|x - x'\|, \; |\langle \nabla g(\bar{x}), x - x'\rangle|\right\}$$

and form the robust distance estimator (Algorithm 1) with $\rho(\cdot, \cdot)$ replacing the Euclidean norm $\|\cdot\|$. In particular, the confidence bound (6.1) and the left inequality in (6.2) imply that $\mathcal{M}(\epsilon)$ is a weak distance oracle, that is, $\mathbb{P}\big(\rho(x, \bar{x}) \leq \epsilon\big) \geq \frac{2}{3}$. A direct extension of

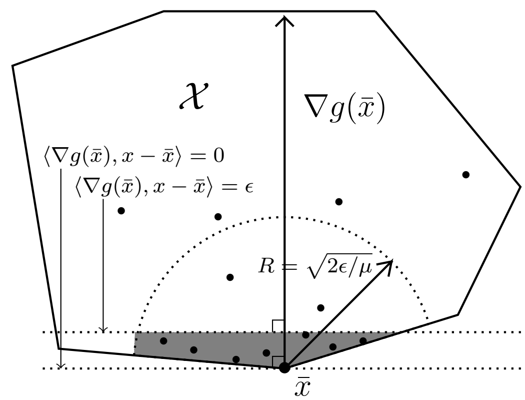

Figure 2: Geometry of the region Λ.

Lemma 2.2 shows that with m calls to the oracle $\mathcal{M}(\epsilon)$, the robust distance estimator returns a point $x \in \mathcal{X}$ satisfying

$$\mathbb{P}\big(\rho(x, \bar{x}) \leq 3\epsilon\big) \geq 1 - \exp(-m/18).$$

Consequently, appealing to the right-hand-side of (6.2), we obtain the desired guarantee

$$\mathbb{P}\big(g(x) - g(\bar{x}) \leq 3(1+\kappa)\epsilon\big) \geq 1 - p.$$

The assumption that we know the gradient $\nabla g(\bar{x})$ is of course unrealistic. Therefore, the strategy we propose will instead replace the gradient $\nabla g(\bar{x})$ with some estimate of the gradient $\nabla g(\hat{x})$ at a nearby point $\hat{x}$, which we denote by $\widetilde{\nabla} g(\hat{x})$. See Figure 3 for an illustration. Indeed, a natural candidate for $\hat{x}$ is the robust distance estimator of $\bar{x}$ in the Euclidean norm. We will see that in order for the proposed procedure to work, it suffices for the gradient estimator $\widetilde{\nabla} g(\hat{x})$ to approximate $\nabla g(\hat{x})$ only up to the very loose accuracy $\kappa\sqrt{\mu\epsilon}$. In particular, if we have access to a stochastic gradient estimator of $\nabla g(\hat{x})$ with variance σ^2, then $\widetilde{\nabla} g$ can be formed using only $\frac{1}{\kappa^2} \cdot \frac{\sigma^2}{\mu\epsilon}$ samples. This overhead in sample complexity is negligible compared to the cost of executing typical algorithms $\mathcal{M}(\epsilon)$, e.g., as given in (5.2) and (5.3).

6.2 Convex composite setting

In this section, we formally develop the procedure that turns low probability guarantees on the function gap for composite problems to high probability outcomes.

Assumption 6.1 (Convex composite problem)**.** We consider the optimization problem

$$\min_{x \in \mathbf{R}^d} \; f(x) := g(x) + h(x) \tag{6.4}$$

where the function $g\colon \mathbf{R}^d \to \mathbf{R}$ is L-smooth and μ-strongly convex and $h\colon \mathbf{R}^d \to \mathbf{R} \cup \{+\infty\}$ is closed and convex. We denote the minimizer of f by $\bar{x}$, its minimal value by $f^* := \min f$, and its condition number by $\kappa := L/\mu$.

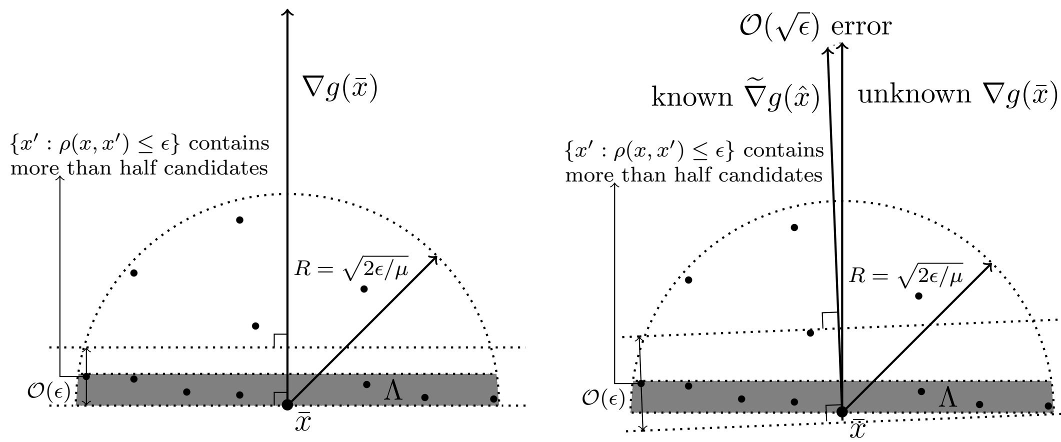

Figure 3: The left side illustrates the region Λ; the right side depicts the perturbation of Λ obtained by replacing the exact gradient $\nabla g(\bar{x})$ with an estimator $\widetilde{\nabla} g(\hat{x}) \approx \nabla g(\hat{x})$.

In particular, we may model minimization of a smooth and strongly convex function g over a closed convex set $\mathcal{X}$ by declaring h to take value zero on $\mathcal{X}$ and $+\infty$ off it. Before stating the proposed algorithm, we require three elementary ingredients: a two-sided bound akin to (2.1), robust distance estimation with a "pseudometric," and a robust gradient estimator.

6.2.1 The two-sided bound

Observe that optimality conditions at $\bar{x}$ imply the inclusion $-\nabla g(\bar{x}) \in \partial h(\bar{x})$, where $\partial h(\bar{x})$ denotes the *subdifferential* of h at $\bar{x}$, and therefore the nonnegativity of the quantity

$$D_h(x, \bar{x}) := h(x) - h(\bar{x}) + \langle \nabla g(\bar{x}), x - \bar{x} \rangle.$$

Indeed, optimization specialists may recognize $D_h(x, \bar{x})$ as a Bregman divergence induced by h—hence the notation. The term $D_h(x, \bar{x})$ appears naturally in a two sided bound similar to (2.1). Specifically, adding $h(x) - h(\bar{x})$ to the two-sided bound (6.2) throughout, we obtain the key two-sided estimate

$$\boxed{D_h(x, \bar{x}) + \frac{\mu}{2}\|x - \bar{x}\|^2 \leq f(x) - f^* \leq D_h(x, \bar{x}) + \frac{L}{2}\|x - \bar{x}\|^2.} \tag{6.5}$$

6.2.2 Robust distance estimation with a pseudometric

As the second ingredient, we will require a slight modification of the robust distance estimation technique of [35, p. 243] and [20]. In particular, it will be convenient to replace the Euclidean norm $\|\cdot\|$ with a more general distance measure.

Definition 6.2 (Pseudometric)**.** A mapping $\rho \colon \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a *pseudometric* on a set $\mathcal{X}$ if for all $x, y, z \in \mathcal{X}$ it satisfies:

1. (nonnegative) $\rho(x, y) \geq 0$ and $\rho(x, x) = 0$,

2. (symmetry) $\rho(x,y) = \rho(y,x)$,

3. (triangle inequality) $\rho(x,y) \le \rho(x,z) + \rho(z,y)$.

The symbol $B_r^\rho(x) = \{y \in \mathcal{X} : \rho(x,y) \le r\}$ will denote the r-radius ball around x in the pseudometric ρ.

With this notation, we record Algorithm 8, which is in the same spirit as the robust distance estimator, Algorithm 1. The differences are that the Euclidean norm is replaced with a pseudometric ρ, an index set is returned instead of a single point, and we leave the origin of the vectors y_i unspecified for the moment.

Algorithm 8: $\texttt{Extract}(\{y_i\}_{i=1}^m, \rho)$
Input: A set of m points $Y = \{y_1, ..., y_m\} \subset \mathcal{X}$, a pseudometric ρ on $\mathcal{X}$. **Step** $i = 1, \ldots, m$: Compute $r_i = \min\{r \ge 0 : \lvert B_r^\rho(y_i) \cap Y\rvert > \frac{m}{2}\}$. Compute the median $\hat{r} = \texttt{median}(r_1, \ldots, r_m)$. **Return** $\mathcal{I} = \{i \in [1,m] : r_i \le \hat{r}\}$.

We will need the following elementary lemma, akin to Lemma 2.2. The main difference is that the lemma provides at least $m/2$ points, instead of a single point, that are close to the target with high probability. The proof is identical to that of [35, p. 243] and [20, Propositions 8 and 9]; we provide details for the sake of completeness.

Lemma 6.3 (Robust distance estimation)**.** *Let ρ be a pseudometric on a set $\mathcal{X}$. Consider a set of points $Y = \{y_1, \ldots, y_m\} \subset \mathcal{X}$ and a point $\bar{y} \in \mathcal{X}$ satisfying $|B_\varepsilon^\rho(\bar{y}) \cap Y| > \frac{m}{2}$ for some $\varepsilon > 0$. Then the index set $\mathcal{I} = \texttt{Extract}(\{y_i\}_{i=1}^m, \rho)$ satisfies the guarantee*

$$\rho(y_i, \bar{y}) \le 3\varepsilon \quad \textit{for all } i \in \mathcal{I}.$$

Proof. Note that for any points $y_i, y_j \in B_\varepsilon^\rho(\bar{y})$, the triangle inequality implies the estimate

$$\rho(y_i, y_j) \le \rho(y_i, \bar{y}) + \rho(\bar{y}, y_j) \le 2\varepsilon.$$

This means that any point $y_i \in B_\varepsilon^\rho(\bar{y})$, at least $\frac{m}{2}$ of them, satisfies $|B_{2\varepsilon}^\rho(y_i) \cap Y| > \frac{m}{2}$ and consequently $r_i \le 2\varepsilon$. Therefore the inequality $\hat{r} = \texttt{median}(r_1, \ldots, r_m) \le 2\epsilon$ holds.

Fix an index $i \in \mathcal{I}$. Since both $B_{r_i}^\rho(y_i)$ and $B_\varepsilon^\rho(\bar{y})$ contain a strict majority of the points in Y, there must exist some point in the intersection $y \in B_{r_i}^\rho(y_i) \cap B_\varepsilon^\rho(\bar{y})$. Using the triangle inequality, we conclude $\rho(y_i, \bar{y}) \le \rho(y_i, y) + \rho(y, \bar{y}) \le 3\varepsilon$, thereby completing the proof. $\square$

6.2.3 Robust gradient estimator

The need for this last ingredient is explained at the end of Section 6.1. Namely, we will need to estimate the gradient $\nabla g(\bar{x})$, thereby perturbing the term $D_h(x,\bar{x})$ in the two-sided bound (6.2). For this purpose, we make the following mild assumption that is standard in applications. Indeed, we already encountered this assumption when paring `proxBoost` with stochastic gradient methods for unconstrained optimization.

Assumption 6.4 (Stochastic first-order oracle)**.** Fix a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ and let $G\colon \mathbf{R}^d \times \Omega \to \mathbf{R}$ be a measurable map satisfying

$$\mathbb{E}_z G(x,z) = \nabla g(x) \qquad \text{and} \qquad \mathbb{E}_z \|G(x,z) - \nabla g(x)\|^2 \leq \sigma^2.$$

We suppose that for any point x, we may sample $z \in \Omega$ and compute the vector $G(x,z)$, which serves as an unbiased estimator of the gradient $\nabla g(x)$.

Under this assumption, we can define a *weak gradient oracle* $\mathcal{G}_\sigma(\cdot,\varepsilon)$ as the averge of a finite sample of stochastic gradients, i.e., for any $\hat{x} \in \mathcal{X}$,

$$\mathcal{G}_\sigma(\hat{x},\varepsilon) := \frac{1}{s}\sum_{i=1}^{s} G(\hat{x}, z_i) \qquad \text{where} \quad s = \left\lceil \frac{3\sigma^2}{\varepsilon^2} \right\rceil.$$

Taking into account the variance reduction by a factor of s and using Markov's inequality, we have

$$\mathbb{P}\left(\left\|\frac{1}{s}\sum_{i=1}^{s} G(\hat{x}, z_i) - \nabla g(\hat{x})\right\|^2 \geq \varepsilon^2\right) \leq \frac{\sigma^2/s}{\varepsilon^2} \leq \frac{1}{3}.$$

That is, $\mathbb{P}\big(\|\mathcal{G}_\sigma(\hat{x},\varepsilon) - \nabla g(\hat{x})\| < \varepsilon\big) \geq \frac{2}{3}$, confirming that $\mathcal{G}_\sigma(\cdot,\varepsilon)$ is indeed a weak distance oracle in the sense of (2.3). Based on this oracle, we can use Algorithm 1 to construct a *robust gradient estimator* $\mathcal{G}_\sigma(\cdot,\varepsilon,m)$, which returns an estimate $\widetilde{\nabla} g(\cdot)$. By Lemma 2.2,

$$\mathbb{P}\big(\|\widetilde{\nabla} g(\hat{x}) - \nabla g(\hat{x})\| \leq 3\varepsilon\big) \geq 1 - \exp(-m/18). \tag{6.6}$$

6.2.4 Robust function gap estimation

Equipped with the three ingredients described above, we present in Algorithm 9 a procedure to robustly estimate the gap $f(x) - f^*$.

Algorithm 9: `RobustGap`$(\mathcal{M}_f(\cdot), m, \epsilon)$.

Input: Minimization oracle $\mathcal{M}_f(\cdot)$, integer $m \in \mathbb{N}$, accuracy $\epsilon > 0$.

Step 1: Independently generate $x_1, ..., x_m$ by the oracle $\mathcal{M}_f(\epsilon)$ so that

$$\mathbb{P}\big(f(x_i) - f^* \le \epsilon\big) \ge \frac{2}{3}, \qquad \text{for all } i \in [1, m]. \tag{6.7}$$

Step 2: Set $\rho_1 = \|\cdot\|$ to be the usual Euclidean norm and compute

$$\mathcal{I}_1 := \texttt{Extract}\left(\{x_i\}_{i=1}^m,\, \rho_1\right). \tag{6.8}$$

Step 3: Fix arbitrary $i \in \mathcal{I}_1$ and set $\hat{x} := x_i$. Use the robust gradient estimator to generate

$$\widetilde{\nabla} g(\hat{x}) := \mathcal{G}_\sigma(\hat{x}, \kappa\sqrt{\mu\epsilon},\, m).$$

Step 4: Define the pseudometric $\rho_2(x, x') := |h(x) - h(x') + \langle \widetilde{\nabla} g(\hat{x}), x - x' \rangle|$ on $\operatorname{dom} h$ and compute

$$\mathcal{I}_2 = \texttt{Extract}(\{x_t\}_{t=1}^m,\, \rho_2).$$

Return: x_i for an arbitrary $i \in \mathcal{I}_1 \cap \mathcal{I}_2$.

Thus, the first step of `RobustGap`$(\mathcal{M}_f(\cdot), m, \epsilon)$ generates m statistically independent points $x_1, \ldots, x_m$ satisfying (6.7). The second step determines a set of points $\{x_i\}_{i \in \mathcal{I}}$ that are all close to $\bar{x}$ with high probability. We then choose a distinguished point $\hat{x} := x_i$ for an arbitrary $i \in \mathcal{I}_1$, and estimate the gradient $\nabla g(\hat{x})$ with $\widetilde{\nabla} g(\hat{x})$. The next step approximates $D_h(\cdot, \bar{x})$ with a pseudometric ρ_2 by replacing $\nabla g(\bar{x})$ with $\widetilde{\nabla} g(\hat{x})$, and then performs robust distance estimation to find a set of points $\{x_i\}_{i \in \mathcal{I}_2}$ with low value of $D_h(x_i, \bar{x})$. Finally a point x_i is returned, for any $i \in \mathcal{I}_1 \cap \mathcal{I}_2$. The intuition is that this x_i simultaneously achieves low values of $\|x_i - \bar{x}\|$ and $D_h(x_i, \bar{x})$, thus allowing us to use (6.5) for robust gap estimation.

The following theorem summarizes the guarantees of the `RobustGap` procedure.

Theorem 6.5 (Robust function gap estimation)**.** *With probability at least* $1 - 2\exp\left(-\frac{m}{18}\right)$, *the point* $x =$ `RobustGap`$(\mathcal{M}_f(\cdot), m, \epsilon)$ *satisfies the guarantee*

$$\|x - \bar{x}\| \le 3\sqrt{\frac{2\epsilon}{\mu}}, \qquad D_h(x, \bar{x}) \le 65\kappa\epsilon, \qquad f(x) - f^* \le 74\kappa\epsilon.$$

In total, the procedure queries m times the oracle $\mathcal{M}_f(\epsilon)$ and evaluates $m \cdot \left\lceil \frac{3\sigma^2}{\kappa^2\mu\epsilon} \right\rceil$ times the stochastic gradient oracle $G(\hat{x}, \cdot)$.

Proof. Define the index set $\mathcal{J} = \{i \in [1, m] : f(x_i) - f^* \leq \epsilon\}$ and define the event

$$E := \left\{|\mathcal{J}| > \frac{m}{2}\right\}.$$

Hoeffding's inequality for Bernoulli random variables guarantees

$$\mathbb{P}(E) \geq 1 - \exp(-m/18).$$

Moreover, using the left inequality in (6.5), we deduce

$$\|x_i - \bar{x}\| \leq \sqrt{\frac{2\epsilon}{\mu}} \quad \text{and} \quad D_h(x_i, \bar{x}) \leq \epsilon \qquad \text{for all } i \in \mathcal{J}. \tag{6.9}$$

Henceforth, suppose that the event E occurs. Then Lemma 6.3 implies

$$\|x_i - \bar{x}\| \leq 3\sqrt{\frac{2\epsilon}{\mu}} \qquad \text{for all } i \in \mathcal{I}_1. \tag{6.10}$$

As discussed in Section 6.2.3, specifically (6.6), the estimate $\widetilde{\nabla} g(\hat{x})$ generated by the robust gradient estimator $\mathcal{G}_\sigma(\hat{x}, \kappa\sqrt{\mu\epsilon}, m)$ satisfies

$$\mathbb{P}(\|\widetilde{\nabla} g - \nabla g(\hat{x})\| \leq 3\kappa\sqrt{\mu\epsilon} \ \mid \ E) \geq 1 - \exp(-m/18). \tag{6.11}$$

Define the event $\hat{E} := \{\|\widetilde{\nabla} g(\hat{x}) - \nabla g(\hat{x})\| \leq 3\kappa\sqrt{\mu\epsilon}\}$ and suppose that $E \cap \hat{E}$ occurs. Then, we compute

$$\begin{aligned}
\|\widetilde{\nabla} g(\hat{x}) - \nabla g(\bar{x})\| &\leq \|\widetilde{\nabla} g(\hat{x}) - \nabla g(\hat{x})\| + \|\nabla g(\hat{x}) - \nabla g(\bar{x})\| \\
&\leq 3\kappa\sqrt{\mu\epsilon} + L\|\hat{x} - \bar{x}\| && (6.12) \\
&\leq 3\kappa\sqrt{\mu\epsilon} + 3L\sqrt{2\epsilon/\mu} = 3(1 + \sqrt{2})\kappa\sqrt{\mu\epsilon}, && (6.13)
\end{aligned}$$

where (6.12) follows from (6.11) and Lipschitz continuity of ∇g, while (6.13) follows from (6.9). Consequently, for each index $i \in \mathcal{J}$, we successively deduce

$$\begin{aligned}
\rho_2(x_i, \bar{x}) &= |h(x_i) - h(\bar{x}) + \langle \widetilde{\nabla} g(\hat{x}), x_i - \bar{x}\rangle| \\
&\leq D_h(x_i, \bar{x}) + |\langle \widetilde{\nabla} g(\hat{x}) - \nabla g(\bar{x}), x_i - \bar{x}\rangle| \\
&\leq \epsilon + 3(1 + \sqrt{2})\kappa\sqrt{\mu\epsilon} \cdot \sqrt{2\epsilon/\mu} && (6.14) \\
&= (1 + (3\sqrt{2} + 6)\kappa)\epsilon,
\end{aligned}$$

where (6.14) follows from (6.9) and (6.13). Therefore, appealing to Lemma 2.2 in the event $E \cap \hat{E}$, we conclude

$$\rho_2(x_i, \bar{x}) \leq 3(1 + (3\sqrt{2} + 6)\kappa)\epsilon \qquad \text{for all } i \in \mathcal{I}_2. \tag{6.15}$$

Finally, fix an arbitrary index $i \in \mathcal{I}_1 \cap \mathcal{I}_2$. We therefore deduce

$$\begin{aligned} D_h(x_i, \bar{x}) &\le \rho_2(x_i, \bar{x}) + |\langle \nabla g(\bar{x}) - \widetilde{\nabla} g, x_i - \bar{x} \rangle| \\ &\le 3(1 + (3\sqrt{2} + 6)\kappa)\epsilon + 3(1 + \sqrt{2})\kappa\sqrt{\mu\epsilon} \cdot 3\sqrt{\frac{2\epsilon}{\mu}} \qquad (6.16) \\ &= 3(1 + (6\sqrt{2} + 12)\kappa)\epsilon \le 65\kappa\epsilon, \qquad (6.17) \end{aligned}$$

where (6.16) follows from the estimates (6.10), (6.13), and (6.15). Using the right side of (6.5), we therefore conclude

$$f(x_i) - f(\bar{x}) \le D_h(x_i, \bar{x}) + \frac{L}{2}\|x_i - \bar{x}\|^2 \le 65\kappa\epsilon + 9\kappa\epsilon = 74\kappa\epsilon,$$

where the last inequality follows from the estimates (6.10) and (6.17). Noting

$$\mathbb{P}(E \cap \hat{E}) = \mathbb{P}(\hat{E} \mid E)\mathbb{P}(E) \ge \left(1 - \exp\left(-\tfrac{m}{18}\right)\right)\left(1 - \exp\left(-\tfrac{m}{18}\right)\right) \ge 1 - 2\exp\left(-\tfrac{m}{18}\right),$$

completes the proof. □

With Theorem 6.5 at hand, we can now replace robust distance estimation with `RobustGap` within the `proxBoost` framework, thereby making `proxBoost` applicable to convex composite problems. The following two sections illustrate the consequences of the resulting method for regularized empirical risk minimization and (proximal) stochastic approximation algorithms.

6.3 Consequences for empirical risk minimization

In this section, we explore the consequences of `RobustGap` and `proxBoost` for regularized empirical risk minimization. In particular, we will boost the low-probability guarantees developed in the seminal work [42] for strongly convex problems to high confidence outcomes. The following assumption summarizes the setting of this section.

Assumption 6.6. Fix a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ and equip $\mathbf{R}^d$ with the Borel σ-algebra. Throughout, we consider the optimization problem

$$\min_x \; f(x) := g(x) + h(x) \qquad \text{where} \qquad g(x) = \mathbb{E}_{z \sim \mathcal{P}}[g(x, z)],$$

under the following assumptions.

1. **(Measurability)** The function $g\colon \mathbf{R}^d \times \Omega \to \mathbf{R}$ is measurable.

2. **(Strong convexity)** The function $h\colon \mathbf{R}^d \to \mathbf{R} \cup \{+\infty\}$ is convex and there exists $\mu > 0$ such that the function $g(x, z) + h(x)$ is μ-strongly convex for a.e. $z \sim \mathcal{P}$.

3. **(Lipschitz continuity)** There exists a measurable map $\ell\colon \Omega \to \mathbf{R}$ and a real $\bar{\ell} > 0$ satisfying the moment bound $\sqrt{\mathbb{E}_z \ell(z)^2} \le \bar{\ell}$ and the Lipschitz condition

$$|g(x, z) - g(y, z)| \le \ell(z)\|x - y\| \qquad \forall x \in U, z \in \Omega,$$

where U is some open neighborhood of dom h.

4. **(Smoothness)** The function $g\colon \mathbf{R}^d \to \mathbf{R}$ is L-smooth.

The first three assumptions are slight modifications of those used in [42], while the additional smoothness assumption on g will be necessary in the sequel to obtain high-confidence guarantees. That being said, the sample efficiency will depend only polylogarithmically on L, and therefore we will be able to treat nonsmooth loss function $g(x,z)$ using standard smoothing techniques. Under the first three assumptions, the authors of [42] obtained the following guarantee for the accuracy of empirical risk minimization.

Lemma 6.7 ([42, Theorem 6])**.** *Let the set $S \subset \Omega$ consist of n i.i.d. samples drawn from $\mathcal{P}$. Then the minimizer of the regularized empirical risk*

$$x_S := \operatorname*{argmin}_x \ \frac{1}{n}\sum_{z\in S} g(x,z) + h(x)$$

satisfies the generalization bound

$$\mathbb{E}_S[f(x_S) - f^*] \leq \frac{2\bar{l}^2}{\mu n}.$$

We will see now how to equip this guarantee with a high confidence bound using `proxBoost`. Recall that to apply the `RobustGap` algorithm, we require an unbiased gradient estimator $G\colon \mathbf{R}^d \times \Omega \to \mathbf{R}^d$ for g. Let us therefore simply declare

$$G(x,z) := \nabla g(x,z).$$

Then we can upper-bound the variance by the second moment

$$\mathbb{E}_z\|G(x,z) - \nabla g(x)\|^2 \leq 2(\mathbb{E}_z\|\nabla g(x,z)\|^2 + \mathbb{E}_z\|\nabla g(x)\|^2) \leq 4\bar{l}^2.$$

We are now ready to present Algorithm 10 as an instantiation of the `proxBoost` procedure for *regularized* empirical risk minimization. In particular, we can still use `ERM`() in Algorithm 3 for the proximal subproblem, with the definition $f(y,z_i) := g(y,z_i) + h(y)$. The robust distance estimator `ERM-R`() in Algorithm 4 can be used without any change.

Algorithm 10: `BoostERMC`(δ, T, m)

Input: accuracy $\delta > 0$, iterations $m, T \in \mathbb{N}$

Set $\lambda_{-1} = 0$, $x_{-1} = 0$

Step $j = 0, \ldots, T$:

  $x_j = \texttt{ERM-R}\left(\frac{54\bar{l}^2}{(\mu+\lambda_{j-1})\delta},\, m,\, \lambda_{j-1},\, x_{j-1}\right)$

Define the minimization oracle $\mathcal{M}_T(\epsilon) := \texttt{ERM}\left(\frac{6\bar{l}^2}{(\mu+\lambda_T)\delta},\, \lambda_T,\, x_T\right)$.

Return $x_{T+1} = \texttt{RobustGap}\left(\mathcal{M}_T\left(\frac{\delta(\mu+\lambda_T)}{222(L+\lambda_T)}\right),\, m,\, \frac{\delta(\mu+\lambda_T)}{222(L+\lambda_T)}\right)$

Notice that in Algorithm 10, we only need to call `RobustGap` in the last cleanup stage, since the intermediate iterations of `proxBoost` only rely on distance estimates to the optimal solutions and not on the function gap. The following theorem and its corollary are immediate consequences of Theorems 3.2 and 6.5.

Theorem 6.8 (Efficiency of `BoostERMC`)**.** *Fix $\delta > 0$ and integers $T, m \in \mathbb{N}$. Then with probability at least $1 - (T+3)\exp\left(-\frac{m}{18}\right)$, the point $x_{T+1} = \texttt{BoostERMC}(\delta, T, m)$ satisfies*

$$f(x_{T+1}) - f^* \leq \left(1 + \sum_{i=0}^{T} \frac{\lambda_i}{\mu + \lambda_{i-1}}\right)\delta.$$

Corollary 6.9 (Efficiency of `BoostERMC` with geometric decay)**.** *Fix a target accuracy $\epsilon > 0$ and a probability of failure $p \in (0,1)$. Define the algorithm parameters:*

$$T = \lceil \log_2(\kappa) \rceil, \qquad m = \left\lceil 18 \ln\left(\frac{T+3}{p}\right)\right\rceil, \qquad \delta = \frac{\epsilon}{4+2T}, \qquad \lambda_i = \mu 2^i.$$

Then the point $x_{T+1} = \texttt{BoostERMC}(\delta, T, m)$ satisfies

$$\mathbb{P}(f(x^{T+1}) - f(x^*) \leq \epsilon) \geq 1 - p.$$

Moreover, the total number of samples used by the algorithm is

$$\mathcal{O}\left(\ln^2(\kappa)\ln\left(\frac{\ln(\kappa)}{p}\right) \cdot \frac{\bar{\ell}^2}{\epsilon\mu}\right). \tag{6.18}$$

Thus, `proxBoost` endows regularized empirical risk minimization with high confidence guarantees at an overhead cost that is only polylogarithmic in κ and logarithmic in $1/p$. In particular, observe that the sample complexity (6.18) established in Corollary 6.9 depends on the smoothness parameter L (through $\kappa = L/\mu$) only polylogarithmically. Consequently, it appears plausible that if the losses $g(\cdot, z)$ are nonsmooth, we may simply replace them by a smooth approximation and apply `BoostERMC`. The price to pay should then only be polylogarithmic in the target accuracy ϵ. Let us formally see how this can be done. To this end, we will assume that the optimization problem in question is to minimize a sum of an expectation of convex functions, a deterministic smooth and strongly convex function (e.g. squared ℓ_2 norm), and a nonsmooth regularizer.

Assumption 6.10. Consider the optimization problem

$$\min_{x} \; f(x) := \mathbb{E}_{z\sim\mathcal{P}}[g(x,z)] + \varphi(x) + h(x) \tag{6.19}$$

under the following assumptions.

1. **(Measurability)** The function $g \colon \mathbf{R}^d \times \Omega \to \mathbf{R}$ is measurable and the assignment $x \mapsto g(x,z)$ is convex for a.e. $z \in \Omega$.

2. **(Strong convexity)** The function $h \colon \mathbf{R}^d \to \mathbf{R} \cup \{+\infty\}$ is convex and there exist parameters $\mu, \beta > 0$ such that the function $\varphi \colon \mathbf{R}^d \to \mathbf{R}$ is μ-strongly convex and β-smooth for a.e. $z \in \Omega$.

3. **(Lipschitz continuity)** There exists a measurable map $\ell \colon \Omega \to \mathbf{R}$ and a real $\bar{\ell} > 0$ satisfying the moment bound $\sqrt{\mathbb{E}_z \ell(z)^2} \leq \bar{\ell}$ and the Lipschitz condition

$$|g(x,z) - g(y,z)| \leq \ell(z)\|x - y\| \qquad \forall x \in \mathbf{R}^d, z \in \Omega.$$

The strategy we follow is to simply replace $g(\cdot, z)$ by a smooth approximation and then apply `BoostERMC`. We now make precise what we mean by a smooth approximation. Assumptions of this type are classical in convex optimization; see for example Nesterov [36] and Beck-Teboulle [7].

Assumption 6.11 (Smoothing)**.** Suppose that for any parameter $\epsilon > 0$, there exist measurable functions $g_\epsilon \colon \mathbf{R}^d \times \Omega \to \mathbf{R}$ and $\ell_\epsilon, L_\epsilon \colon \Omega \to \mathbf{R}_+$ such that $g_\epsilon(\cdot, z)$ is Lipschitz continuous with constant $\ell_\epsilon(z)$ and its gradient is Lipschitz continuous with constant $L_\epsilon(z)$, and the estimate holds:

$$|g(x,z) - g_\epsilon(x,z)| \leq \epsilon \qquad \text{for all } x \in \mathbf{R}^d, z \in \Omega. \tag{6.20}$$

We suppose moreover that the moment conditions, $\sqrt{\mathbb{E}_z \ell_\epsilon^2(z)} \leq \bar{\ell}_\epsilon$ and $\mathbb{E}_z L_\epsilon(z) \leq \bar{L}_\epsilon$, hold for some constants $\bar{\ell}_\epsilon, \bar{L}_\epsilon > 0$.

Let us look at two standard examples of smoothings of convex functions.

Example 6.1 (Moreau envelope)**.** A classical approach to smoothing a convex function is based on the Moreau envelope [33]. Namely, fix a convex function $\psi \colon \mathbf{R}^d \to \mathbf{R}$. The Moreau envelope of ψ with parameter $\nu > 0$ is defined to be

$$M_\nu^\psi(x) = \min_y \ \psi(y) + \frac{1}{2\nu}\|y - x\|^2.$$

It is well-known that M_ν^ψ is $\frac{1}{\nu}$-smooth. Moreover if ψ is Lipschitz continuous with constant $\mathrm{lip}\,(\psi)$, then M_ν^ψ is also Lipschitz continuous with the same constant and the bound holds:

$$0 \leq \psi(x) - \psi_\nu(x) \leq \nu(\mathrm{lip}\,(\psi))^2.$$

Coming back to our target problem (6.19), we may define $g_\epsilon(\cdot, z)$ to be the Moreau envelope of $g(\cdot, z)$ with parameter $\nu(z) := \frac{\epsilon}{l(z)^2}$, or more explicitly

$$g_\epsilon(x,z) = \min_y \ g(y,z) + \frac{l(z)^2}{2\epsilon}\|y - x\|^2.$$

Then the parameters from Assumption 6.10 become $\ell_\epsilon(z) := \ell(z)$ and $L_\epsilon(z) := \frac{l(z)^2}{\epsilon}$.

Example 6.2 (Compositional smoothing)**.** Often, the Moreau envelope of $g(\cdot, z)$ may be difficult to compute explicitly. In typical circumstances, however, the function $g(\cdot, z)$ may be written as a composition of a simple nonsmooth convex function with a linear map. It then suffices to replace only the outer function with its Moreau envelope—a technique famously explored by Nesterov [36].

To illustrate on a concrete example, suppose that the population data consists of tuples $z = (a, b) \sim \mathcal{P}$ and the loss takes the form $g(x, z) = h(\langle a, x\rangle, b)$ for some measurable function $h(\cdot, \cdot)$ that is convex and 1-Lipschitz in its first argument. In order to control the Lipschitz constant, suppose also the moment bound $\sqrt{\mathbb{E}_a\|a\|^2} \leq A$ for some constant $A > 0$. Let us now define the smoothing

$$g_\epsilon(x, z) = h_\epsilon(\langle a, x\rangle, b),$$

where $h_\epsilon(\cdot, b)$ is the Moreau envelope of $h(\cdot, b)$ with parameter $\nu = \epsilon$. It is straightforward to verify that the estimate (6.20) holds and that we may set $\ell_\epsilon(z) = A$ and $L_\epsilon(z) = \frac{A^2}{\epsilon}$.

With Assumptions 6.10 and 6.11 at hand, we may now simply apply `BoostERMC` to the smoothed problem

$$\min_x \; f_\epsilon(x) := g(x) + h(x) \qquad \text{where} \qquad g(x) = \mathbb{E}_{z\sim\mathcal{P}}[g_\epsilon(x,z)] + \varphi(x).$$

Using Corollary 6.9, we deduce that the procedure will find a point x satisfying

$$\mathbb{P}(f_\epsilon(x) - f_\epsilon^* \le \epsilon) \ge 1 - p,$$

using

$$\mathcal{O}\left(\ln^2\left(\tfrac{\bar{L}_\epsilon+\beta}{\mu}\right)\ln\left(\tfrac{\ln(\frac{\bar{L}_\epsilon+\beta}{\mu})}{p}\right)\cdot\frac{\bar{\ell}_\epsilon^2}{\epsilon\mu}\right)$$

samples. Observe that with probability $1-p$ the returned point x satisfies:

$$f(x) - f^* \le f_\epsilon(x) - f_\epsilon^* + (f(x) - f_\epsilon(x)) + (f_\epsilon^* - f^*) \le 3\epsilon.$$

In particular, in the setup of Examples 6.1 and 6.2, the sample complexities become:

$$\mathcal{O}\left(\ln^2\left(\tfrac{\bar{\ell}^2/\epsilon+\beta}{\mu}\right)\ln\left(\tfrac{\ln(\frac{\bar{\ell}^2/\epsilon+\beta}{\mu})}{p}\right)\cdot\frac{\bar{\ell}^2}{\epsilon\mu}\right) \quad \text{and} \quad \mathcal{O}\left(\ln^2\left(\tfrac{A^2/\epsilon+\beta}{\mu}\right)\ln\left(\tfrac{\ln(\frac{A^2/\epsilon+\beta}{\mu})}{p}\right)\cdot\frac{A^2}{\epsilon\mu}\right),$$

respectively. Hence, the price to pay for nonsmoothness is only polylogarithmic in $1/\epsilon$.

6.4 Consequences for stochastic approximation

We now extend the results of Section 5 to the convex composite setting. In addition to Assumptions 6.1 and 6.4, in this section we will use the following composite analogue of Assumption 5.1. At the end of the section, we will let `Alg`$(\cdot)$ be the (accelerated) proximal stochastic gradient method.

Assumption 6.12. Consider the proximal minimization problem

$$\min_y \; \varphi_x(y) := g(y) + \frac{\lambda}{2}\|y-x\|^2 + h(y),$$

Let $\Delta > 0$ be a real number satisfying $\varphi_x(x) - \min \varphi_x \le \Delta$. We will let `Alg`$(\delta, \lambda, \Delta, x)$ be a procedure that returns a point y satisfying

$$\mathbb{P}[\varphi_x(y) - \min \varphi_x \le \delta] \ge \frac{2}{3}.$$

The following algorithm is a direct extension of `BoostAlg` (Algorithm 6) to the convex composite setting; the only difference is that `BoostAlgC` (Algorithm 9) replaces the distance estimator `Alg-R`$(\cdot)$ with `RobustGap`$(\cdot)$.

Algorithm 11: `BoostAlgC`$(\delta, \Delta_{\rm in}, x_{\rm in}, T, m)$

Input: accuracy $\delta > 0$, upper bound $\Delta_{\rm in} > 0$, initial $x_{\rm in} \in \mathbf{R}^d$, numbers $m, T \in \mathbb{N}$
Set $\lambda_{-1} = 0$, $\Delta_{-1} = \Delta_{\rm in}$, $x_{-1} = x_{\rm in}$
Step $j = 0, \ldots, T$:
Define the minimization oracle for the proximal subproblem

$$\mathcal{M}_{j-1}(\cdot) := \texttt{Alg}(\cdot,\, \lambda_{j-1},\, \Delta_{j-1},\, x_{j-1}).$$

Set $x_j = \texttt{RobustGap}\big(\mathcal{M}_{j-1}(\delta/9),\, m,\, \delta/9\big)$
Set $\Delta_j = \delta\left(9 \cdot \frac{L+\lambda_{j-1}}{\mu+\lambda_{j-1}} + \sum_{i=0}^{j-1} \frac{\lambda_i}{\mu+\lambda_{i-1}}\right)$
Return $x_{T+1} = \texttt{RobustGap}\left(\mathcal{M}_T\left(\frac{\delta(\mu+\lambda_T)}{74(L+\lambda_T)}\right),\, m,\, \frac{\delta(\mu+\lambda_T)}{74(L+\lambda_T)}\right)$

In `BoostAlgC`, we need to use `RobustGap` in every proximal iteration as well as the cleanup step, because the stochastic proximal gradient method encoded as `Alg` typically requires robust gap estimation on the initialization gap Δ_j. The proof of the following theorem is almost identical to that of Theorem 5.2, with Theorem 6.5 playing the role of Lemma 2.2.

Theorem 6.13 (Efficiency of `BoostAlgC`)**.** *Fix an arbitrary point $x_{\rm in} \in \mathbf{R}^d$ and let $\Delta_{\rm in}$ be any upper bound $\Delta_{\rm in} \geq f(x_{\rm in}) - \min f$. Fix natural numbers $T, m \in \mathbb{N}$. Then with probability at least $1 - 2(T+2)\exp\left(-\frac{m}{18}\right)$, the point $x_{T+1} =$ `BoostAlgC`$(\delta, \Delta_{\rm in}, x_{\rm in}, T, m)$ satisfies*

$$f(x_{T+1}) - \min f \leq \delta\left(1 + \sum_{i=0}^{T} \frac{\lambda_i}{\mu + \lambda_{i-1}}\right).$$

When using the proximal parameters $\lambda_i = \mu 2^i$, we obtain the following guarantee, which generalizes Corollary 5.3 to the composite setting.

Corollary 6.14 (Efficiency of `BoostAlgC` with geometric decay)**.** *Fix an arbitrary point $x_{\rm in} \in \mathbf{R}^d$ and let $\Delta_{\rm in}$ be any upper bound $\Delta_{\rm in} \geq f(x_{\rm in}) - \min f$. Fix a target accuracy $\epsilon > 0$ and probability of failure $p \in (0,1)$, and set the algorithm parameters*

$$T = \lceil \log_2(\kappa) \rceil, \qquad m = \left\lceil 18 \ln\left(\frac{4+2T}{p}\right)\right\rceil, \qquad \delta = \frac{\epsilon}{2+2T}, \qquad \lambda_i = \mu 2^i.$$

Then the point $x_{T+1} =$ `BoostAlg`$(\delta, \Delta_{\rm in}, x_{\rm in}, T, m)$ satisfies

$$\mathbb{P}(f(x_{T+1}) - \min f \leq \epsilon) \geq 1 - p.$$

Moreover, the total number of calls to `Alg`$(\cdot)$ is

$$\left\lceil 18 \ln\left(\frac{4 + 2\lceil \log_2(\kappa)\rceil}{p}\right)\right\rceil \lceil 2 + \log_2(\kappa)\rceil,$$

the number of evaluations of the stochastic gradient oracle $G(\cdot,\cdot)$ is at most[1]

$$\left\lceil 18\ln\left(\frac{\lceil 4+2\log_2(\kappa)\rceil}{p}\right)\right\rceil \cdot \left\lceil \frac{6\sigma^2}{L\epsilon}\cdot(2+2\lceil\log_2(\kappa)\rceil)\right\rceil \cdot \lceil 2+\log_2(\kappa)\rceil,$$

and the initialization errors satisfy

$$\max_{i=0,\ldots,T+1}\Delta_i \leq \frac{9\kappa+1+2\lceil\log_2(\kappa)\rceil}{2+2\lceil\log_2(\kappa)\rceil}\epsilon.$$

In particular, the stochastic gradient method and its accelerated variant [18,25] admit proximal extensions with exactly the same sample complexities as in the smooth case, (5.2) and (5.3), respectively. Clearly, we may use either of these two procedures as `Alg(·)` within Algorithm 11. Corollary 6.14 then immediately shows that the two resulting algorithms will find a point x satisfying $\mathbb{P}[f(x)-f^*\leq\epsilon]\geq 1-p$ with the same sample complexities as in the smooth setting, (5.4) and (5.5), respectively.

References

[1] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *J. Comput. System Sci.*, 58(1, part 2):137–147, 1999. Twenty-eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996).

[2] H. Asi and J.C. Duchi. Stochastic (approximate) proximal point methods: Convergence, optimality, and adaptivity. *arXiv:1810.05633*, 2018.

[3] H. Asi and J.C. Duchi. The importance of better models in stochastic optimization. *arXiv:1903.08619*, 2019.

[4] Necdet Serhat Aybat, Alireza Fallah, Mert Gurbuzbalaban, and Asuman Ozdaglar. A universally optimal multistage accelerated stochastic gradient method. *arXiv preprint arXiv:1901.08022*, 2019.

[5] P.L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[6] A. Beck. *First-order methods in optimization*, volume 25. SIAM, 2017.

[7] A. Beck and M. Teboulle. Smoothing and first order methods: a unified framework. *SIAM J. Optim.*, 22(2):557–580, 2012.

[8] O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[1]For the middle term, we use the observation, $\min_{\lambda\geq 0}\frac{(L+\lambda)^2}{\mu+\lambda}\geq\frac{L}{2}$, which is straightforward to verify.

[9] S. Bubeck, N. Cesa-Bianchi, and G. Lugosi. Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59(11):7711–7717, 2013.

[10] O. Catoni. Challenging the empirical mean and empirical variance: a deviation study. *Ann. Inst. Henri Poincaré Probab. Stat.*, 48(4):1148–1185, 2012.

[11] Y. Chen, L. Su, and J. Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):44, 2017.

[12] D. Davis and D. Drusvyatskiy. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019.

[13] D. Davis and B. Grimmer. Proximally guided stochastic method for nonsmooth, nonconvex problems. *arXiv:1707.03505*, 2017.

[14] J.C. Duchi and F. Ruan. Stochastic methods for composite and weakly convex optimization problems. *SIAM Journal on Optimization*, 28(4):3229–3259, 2018.

[15] R. Frostig, R. Ge, S. Kakade, and A. Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

[16] R. Ge, S. Kakade, R. Kidambi, and P. Netrapalli. The step decay schedule: A near optimal, geometrically decaying learning rate procedure. *arXiv:1904.12838*, 2019.

[17] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: A generic algorithmic framework. *SIAM J. Optim.*, 22(4):1469–1492, 2012.

[18] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, II: shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 23(4):2061–2089, 2013.

[19] J.L. Goffin. On convergence rates of subgradient optimization methods. *Math. Programming*, 13(3):329–347, 1977.

[20] D. Hsu and S. Sabato. Loss minimization and parameter estimation with heavy tails. *The Journal of Machine Learning Research*, 17(1):543–582, 2016.

[21] M.R. Jerrum, L.G. Valiant, and V.V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoret. Comput. Sci.*, 43(2-3):169–188, 1986.

[22] E. Joly, G. Lugosi, and R.I. Oliveira. On the estimation of the mean of a random vector. *Electronic Journal of Statistics*, 11(1):440–451, 2017.

[23] A. Juditsky, A. Nazin, A. Nemirovsky, and A. Tsybakov. Algorithms of robust stochastic optimization based on mirror descent method. *arXiv:1907.02707*, 2019.

[24] A. Juditsky and Y. Nesterov. Deterministic and stochastic primal-dual subgradient algorithms for uniformly convex minimization. *Stoch. Syst.*, 4(1):44–80, 2014.

[25] A. Kulunchakov and J. Mairal. Estimate sequences for stochastic composite optimization: Variance reduction, acceleration, and robustness to noise. *arXiv:1901.08788*, 2019.

[26] G. Lan and Y. Zhou. An optimal randomized incremental gradient method. *Math. Program.*, 171(1-2, Ser. A):167–215, 2018.

[27] H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pages 3366–3374, 2015.

[28] G. Lugosi and S. Mendelson. Risk minimization by median-of-means tournaments. *arXiv:1608.00757*, 2016.

[29] G. Lugosi and S. Mendelson. Sub-gaussian estimators of the mean of a random vector. *The Annals of Statistics*, 47(2):783–794, 2019.

[30] B. Martinet. Régularisation d'inéquations variationnelles par approximations successives. *Rev. Française Informat. Rech. Opérationnelle*, 4(Sér. R-3):154–158, 1970.

[31] B. Martinet. Détermination approchée d'un point fixe d'une application pseudo-contractante. Cas de l'application prox. *C. R. Acad. Sci. Paris Sér. A-B*, 274:A163–A165, 1972.

[32] S. Minsker. Geometric median and robust estimation in Banach spaces. *Bernoulli*, 21(4):2308–2335, 2015.

[33] J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France*, 93:273–299, 1965.

[34] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.*, 19(4):1574–1609, 2008.

[35] A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization.* A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 1983. Translated from the Russian and with a preface by E. R. Dawson, Wiley-Interscience Series in Discrete Mathematics.

[36] Yu. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1, Ser. A):127–152, 2005.

[37] Yu. Nesterov. *Lectures on convex optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer, 2018.

[38] Yu. Nesterov and J.-P. Vial. Confidence level solutions for stochastic programming. *Automatica*, 44:1559–1568, 2008.

[39] B.T. Polyak and A.B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.*, 30(4):838–855, 1992.

[40] R.T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control Optimization*, 14(5):877–898, 1976.

[41] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[42] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Stochastic convex optimization. In *COLT*, 2009.

[43] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Math. Program.*, 155(1-2, Ser. A):105–145, 2016.

[44] A. Shapiro and A. Nemirovski. On complexity of stochastic programming problmes. In V. Jeyakuman and A. M. Rubinov, editors, *Continuous optimization: Current trends and applications*, pages 111–144. Springer, 2005.

[45] Y. Xu, Q. Lin, and T. Yang. Accelerated stochastic subgradient methods under local error bound condition. *arXiv preprint arXiv:1607.01027*, 2016.

[46] T. Yang, Y. Yan, Z. Yuan, and R. Jin. Why does stagewise training accelerate convergence of testing error over SGD? *arXiv:1812.03934*, 2018.

[47] D. Yin, Y. Chen, R. Kannan, and P. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5650–5659, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR.